DOCTOR OF PHILOSOPHY

Towards a formal ontology to support knowledge sharing across product design and manufacturing

Saha, Sattam

Award date:
2021

Awarding institution:
Coventry University

Towards A Formal Ontology To Support Knowledge Sharing Across Product Design And Manufacturing

By

Sattam Saha

July 2020

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

Content removed on data protection grounds

Ethical Approval

Certificate of Ethical Approval

Applicant:

Sattam Saha

Project Title:

Developing a formal manufacturing ontology to significantly contribute towards seamless knowledge sharing across multiple manufacturing domains such as design, assembly, welding machining and inspection.

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Medium Risk

Date of approval:

06 June 2017

Project Reference Number:

P46315

Candidate Declaration Form

Content removed on data protection grounds

Abstract

The ability to seamlessly share knowledge across different product lifecycle domains is a crucial enabler for decision making. It dictates the competence of a manufacturing enterprise. This transcends over to Information and Communication Technology (ICT) systems, which are increasingly becoming an integral part of design and manufacturing stages. In today's competitive manufacturing world, these systems are required to seamlessly share knowledge for better, faster and cheaper production. However, different manufacturing domains have different data structures and syntaxes leading to knowledge sharing issues. Furthermore, the loosely defined semantic of the contributing concepts and relations lead to different, sometimes contradicting interpretations. Thus, the knowledge sharing capability of such systems across design and manufacturing domains are impeded. A computationally interpretable ontology can resolve these issues by providing a basis for common understanding across these domains.

In this thesis, a unique solution in the form of a Product Lifecycle Ontology (PLO) is proposed that facilitates semantic knowledge sharing across product design and manufacture. The proposed ontology supports this by providing a common semantic base that provides a route to link domains and enable knowledge sharing. The research work demonstrates sharing of knowledge from Machining, Welding and Inspection with Design. This is achieved by defining a set of concepts and relations with rigorous formal semantics. An approach to specialise these concepts at multiple new levels to capture the varying depth of meanings with higher granularity has been presented. This has further been utilised to develop a novel model for classification of joining and welding processes that facilitates reconciliation of international welding standards. Similarly, an innovative way to categorise different types of manufacturing operations and efficiently model their sequences was unveiled.

The ontology is verified experimentally and through and an industrial case study. The research work has shown the potential to reduce the number of design revisions by capturing the manufacturing specific knowledge and share it with product design. Further, the recommendation of this work is ready to be fed into the technical committees overseeing the welding standards to improve them for better interoperability. Thus, the proposed ontology expands previous works and fills in the existing research gaps within the area of formal manufacturing reference ontologies.

Acknowledgement

My journey towards attaining a PhD through this research work involved contributions from many different individuals, directly and indirectly. I would like to extend my gratitude to all of them for their continuous support and contributions. I am forever grateful to you.

Firstly, I would like to express my sincere thanks to my supervisor Dr. Zahid Usman. I am obliged to him for giving me the opportunity to carry out this research. Dr. Zahid was initially my main supervisor before he moved to Rolls-Royce and became my industrial supervisor. His technical guidance has been crucial in enhancing my fundamental understandings of this research. Moreover, his advice from an industrial perspective allowed me to implement a real world validation to this research.

I would like to extend my earnest gratitude towards my Director of Studies, Prof. Weidong Li. He has been an immense support during my research journey. Prof. Weidong is an expert in his field with abundance of the most updated knowledge and his guidance has been enormously valuable. His critical analysis has been crucial for my research. It helped me to delve, improve the finer aspects of the research and develop similar critical thinking skills. As a leading researcher across multiple countries, Prof. Weidong always had time for me even during his busiest schedules. His contributions have been more indispensable with regards to writing and publishing research papers. It has been a privilege to work with him and has helped me develop not only as a researcher but also as a better human being.

I would also like to thank my co-supervisor Dr. Nazaraf Shah for his valuable input, feedback with regards to writing my thesis and research papers. Furthermore, I would like to extend this gratitude to Prof. Steve Jones for his valuable technical insights during my initial days of the research.

My PhD was funded by the Institute For Advanced Manufacturing and Engineering (AME). A part of the Research Centre For Future Transport and Cities (FTC) of Coventry University, I am really grateful to Prof. Carl Perrin and Mr. Nick Turner for this. To the extended staff members of AME, it was an immense pleasure to work with them and would like to thank all. During this time, I was privileged to meet Mrs. Sharwari Pujari, Mr. Rizwan Tai and Ms. Jennifer Quinn who have been a constant support.

I wish to thank everyone at Rolls-Royce Plc who has immensely supported me during my industrial placement. Massive thanks goes to the entire Digital Manufacturing, Manufacturing Engineering and Design Engineering teams of Compressor Components. A special thanks to Dr. Neil Hastilow and Mr. Gavin Morgan for the placement opportunity. I would like to specially mention Ms. Bethan Murray, Mr. Michael Ball, Mr. Dan Swan, Mr. David Barmby, Mr. Guillaume Delannoy, Ms. Lori Liu and Dr. Gervasio Salerno for their constant support during my industrial placement.

Finally, an immense appreciation goes to my family, friends and teachers. I would like to specially mention my friend Mr. Antariksh Akhave who helped me to secure this opportunity. I would like to dedicate this work to my father and mother who have been the relentless sources of motivation. They have been with me through thick and thin. Thank You for believing in me. I am here because of you.

Publications

1. **Saha S**, Usman Z, Li Wei, Jones S, Kshirsagar R, "**A Formal Ontological Approach for Semantic Coherence in Welding Standards.**" In Advances in Manufacturing Technology XXXI, edited by J. Gao et al., 503-508. London: IOS Press, 2017.
2. **Saha S**, Usman Z, Li Wei, Jones S, Kshirsagar R, "**Towards A Formal Ontology To Support Interoperability Across Multiple Product Lifecycle Domains** in the Proceedings of IEEE 11th International Conference on Semantic Computing." San Diego, CA: IEEE Xplore, 2017.
3. **Saha. S**, Usman Z, Li Wei, Jones S, Shah N, "**Core domain ontology for joining processes to consolidate welding standards.**" Robotics and Computer Integrated Manufacturing 59 (2019): 417-430.
4. **Saha. S**, Li Wei, Usman Z, Shah N, "**Core Manufacturing Ontology to Model Manufacturing Operations and Sequencing Knowledge**." Computers and Industrial Engineering. *(Submitted: Under Review)*

Table of Contents

List of Figures

List of Tables

Abbreviations

ADACOR – ADAptive holonic COntrol aRchitecture

AIF – ATHENA Interoperability Framework

ATHENA – Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications

BFO – Basic Formal Ontology

CAD – Computer Aided Design

CAE – Computer Aided Engineering

CAM – Computer Aided Manufacturing

CAPP – Computer Aided Process Planning

CAx – Computer Aided Technologies

CIM – Computer Independent Model

CIMOSA – Computer Integrated Manufacturing Open System Architecture

CL – Common Logic

CLIF – Common Logic Interchange Format

CPM – Core Product Model

DL – Description Logic

DOLCE – Descriptive Ontology for Linguistic and Cognitive Engineering

ERP – Enterprise Resource Planning

FOL – First Order Logic

GNP – Gross National Product

GDP – Gross Domestic Product

ICT – Information and Communications Technology

ISO – International Organisation for Standardization

KB – Knowledge Base

KBE – Knowledge Based Engineering

KFL Knowledge Frame Language

KIF Knowledge Interchange Format

KM Knowledge Management

KMS Knowledge Management Systems

MAFRA ontology MApping FRAmework

MANDATE – MANufacturing management DATa interchange

MASON – Manufacturing Semantics ONtology

MDA – Model Driven Architecture

MDI – Model Driven Interoperability

MRP – Material Resource Planning

MSE – Manufacturing System Engineering

NIST – National Institute of Standards and Technology

OCHRE – Object Centred High-level REference ontology

OWL – Web Ontology Language

PERA – Purdue Enterprise Reference Architecture

PIM – Platform Independent Model

PSM – Platform Specific Model

PLIB – Parts LIBrary

PLM – Product Lifecycle Management

PLO – Product Lifecycle Ontology

PSL – Process Specification Language

PSRL – Product Semantic Representation Language

RDF – Resource Description Framework

SMIF – Semantic Manufacturing Interoperability Framework

STEP – Standard for the Exchange of Product model data

SWRL – Semantic Web Rule Language

UML – Unified Modelling Language

URI – Uniform Resource Identifier

W3C – World Wide Web Consortium

XML – eXtensible Markup Language

1. Introduction

1.1 Research Background

Manufacturing industry is one of the key influencers of world economy. It dictates the socio economic condition as it is the largest employment sector. This plays a pivotal role in determining national wealth, power and prosperity (Rynn, 2011) (Manufacturer, 2018). In order to be competitive, manufacturers constantly strive to produce better quality products at lower costs within shortest amount of time. A "right first time" approach is required to reduce the several cycles of revisions. This is dependent on the informed decision making capability of the designers, manufacturing engineers and production planners. Predominantly, it is the design engineers who are required to be aware of the consequences of their design on the manufacturing activities such as machining, assembly, welding and inspection. Thus, seamless exchange of product lifecycle knowledge is of paramount importance.

Concurrent engineering is one of the approaches which are generally employed during product development for exchange of knowledge (Loureiro., 2018). This entails the design engineers and the experts from different manufacturing domains to work collaboratively in unison and requires seamless interaction across the domains. Such an approach works well in human centric environment as any semantic contradictions can be resolved through human interactions. However, it is a challenge when different Information and Communication Technology (ICT) systems are required to share knowledge and resolve any variances. The emergence of ICT based systems have resulted in manufacturing organisations utilising them as a supporting tool for the purpose of information capture and sharing. But today's manufacturing organisations are large and encompass various discrete domains. Thus they require multiple systems. In general, these large multinational manufacturing enterprises manage their product lifecycle knowledge using multiple different Knowledge Management Systems (KMS) and tools. Some of these are

1. Enterprise Resource Planning (ERP) Systems and Material Resource Planning (MRP) Systems such as SAP,
2. Product Lifecycle Management (PLM) Systems and,

3. Manufacturing Execution Systems (MES) for product definitions and process planning.

It is imperative that these systems are able to communicate and seamlessly exchange knowledge for the organisation to be efficient. The integration of the multiple enterprise systems acts as a barrier (Hastilow, 2013).These systems are flawless independently but have limited capability for knowledge sharing (Peng., 2020). It is due to their incapability of representing and sharing information seamlessly (Young., 2010) (Hedberg., 2016) . This is where "interoperability" comes into play, which essentially is the ability to seamlessly exchange information across systems (IEEE-Std-Computer-Dictionary, 1991) (ISO/IEC-TR-10000-3., 1998) (Mourad., 2016) . It has been perceived that addressing the interoperability issues would contribute massively towards seamless knowledge sharing across multiple enterprise domains (Imran, 2013) (Usman, 2012) (Liu., 2020)

The majority of the interoperability problems originates from the information being structurally held on diversified product and manufacturing models (Srinivasan, 2011) . This is highly predominant in large multi-national organisations which tend to work in silos, making it more challenging and expensive. Further, the lack of formalisation (computer interpretable logics), variety of data structures and formats, incoherent semantics of concepts and syntaxes across multiple product lifecycle domains leads to interoperability problems.

Interoperability problems have been estimated to cost about $1 billion annually to the automotive sector of the United States (US) (Brunnermeier, 2002) and $15.8 billion to their capital facilities (Gallaher, 2009). It has been reported that about $31.5 billion is spent annually by the Fortune 500 companies to overcome interoperability problems (Babcock, 2004). Among the interoperability issues, about 70% of the interoperability costs have been reported to be spent on reconciliation of semantic inconsistencies (Bussler, 2005) (Ahmed., 2013) . Therefore, it is essential for product and manufacturing models to be devoid of semantic inconsistencies. It is a fundamental requirement for consistent knowledge capture and sharing among different application domains to ensure interoperability. Semantic inconsistencies pertain to the differences in the meaning of concepts. One of the potential ways to overcome interoperability problems is through standards as they are perceived to be the agreed global references to support the wider industrial requirements. Several international standards have been developed for product data and information management to aid interoperability. However these standards have been reported to have semantic inconsistencies (Chungooraa et al, 2013). Further, the concepts in these standards were found to have subjective interpretation (Michel, 2005) (Gunendran et al, 2007) (Young et al., 2007). Thus, standards are inefficient and error-prone for interoperability across ICT based systems.

The interoperability issues can be resolved through a common semantic base that resolves and reconciles any sematic mismatches (Hakimpour, 2003) (Chen, 2004) (Fotineau., 2013) . Ontologies provide such a semantic base for proficient knowledge management as they are able to explicitly represent and exchange data semantics (Qin., 2018). There are several definitions of ontology found in literature (Gruber, 1993) (Gruninger., 1996) (Guarino, 1997) (McGuinness, 2002) (Blomqvist, 2008) (Mourad., 2016), but the one most relevant to this work is “a lexicon of the specialized terminology along with some specifications of the meanings of the terms involved” (ISO 18629-1, 2004). This definition paves its way into the different types of ontologies through the usage of the phrase “some specification of the meanings”. Ontologies can be broadly divided into lightweight (informal) and heavyweight (formal) ontologies (Go´mez-Pe´rez, 2004). The latter is computer interpretable through the use of constraints for restricting the meaning of the terms. Formal ontologies also utilise inference rules to deduce new knowledge from existing knowledge (Imran, 2013) (Peng., 2020). Lightweight ontologies on the other hand comprise of simple taxonomies of concepts which are open to differential and erroneous interpretation (Young et al., 2007). Hence, formal ontologies have been found to provide a more rigorous sematic base that can facilitate knowledge sharing across the domains of design and manufacturing. Although the heavyweight or formal ontologies are capable of overcoming their drawbacks but its utilisation in the manufacturing domain has been limited. This can be observed within the current KMS, as their underlying structures are based on different standards (Ray, 2006) or lightweight ontologies. A generic product lifecycle ontology that can act as a semantic base for multiple domains across design and manufacturing has been impending. In this thesis, a formal Product Lifecycle Ontology (PLO) is proposed that can provide a semantic base for knowledge sharing across design, machining, welding and inspection domains. The proposed ontology comprises of core concepts, relations and axioms for multiple manufacturing domains. It provides an additional capability to reconcile diverse set of concepts and deduce new knowledge.

1.2 Hypothesis

The hypothesis for this thesis is that

"A formal core ontology can support knowledge sharing from machining, welding and inspection domains with product design by providing a common verifiable semantic base."

The verification of this hypothesis involves the creation of a generic ontology comprising of an extensive collection of core concepts from different domains of design and manufacturing. Figure 1 gives an overview of the proposed research hypothesis. The product lifecycle core ontology shown in the Figure 1 is formed by formally defining the core set of concepts and their relations for multiple domains.

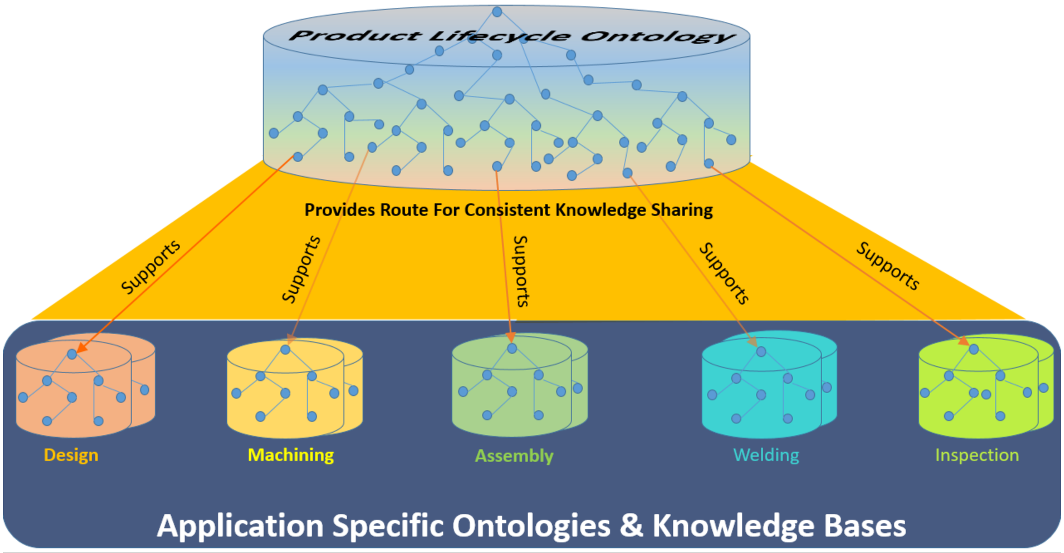


Figure 1Overview of proposed research hypothesis

It's natural for concepts belonging to one particular domain to have a different implications and perspectives across other domains. Thus, an intermediate set of concepts are required which are capable of capturing relatable knowledge and make them understandable across other domains. In this research, these intermediate set of concepts are termed as "core" concepts. They are more generic concepts than those specific for individual domains of design, machining, assembly with welding and inspection. These concepts are to be formalised using computer interpretable logics to ensure that they are unambiguous. Further, a commitment to this ontology would guarantee the consistency of the knowledge being captured. Therefore it ensures seamless exchange of knowledge between manufacturing and product design. The commitment to the proposed ontology is attained by specialising the core concepts into the different domain specific concepts.

1.3 Aims and Objectives

The aim of this research is to contribute and enhance the understanding of formal ontological approaches in developing a semantic base that allows knowledge sharing from machining, welding and inspection domains with product design.

The research hypothesis in the previous section is utilised to achieve this aim. Its successful accomplishment contributes towards better understanding of formal ontology's proficiency to seamlessly capture and share the manufacturing knowledge with product design. Further, it improves the knowledge regarding utilising ontology based decision support systems for PLM. The proposed aim is perceived to be achieved by developing a core ontology that would support the development of domain specific ontologies for different design and manufacturing domains. These, in turn would support the development of application specific ontologies. Thus, the core ontology would act as the route for knowledge sharing and reuse.

The following objectives have to be met to achieve aim of this research.

1. To propose an ontological framework and identify the core concepts, their semantics and relationships for improved knowledge sharing across design, machining, welding and inspection.
2. To formally define the identified concepts and relations such that the semantic inconsistencies and misinterpretations are removed for intra and inter domain knowledge sharing.
3. To explicitly capture the varying depth of meanings of the concepts at different levels and reconcile the semantics of divergent concepts.
4. To develop a core ontological model for joining processes and consolidate the welding standards.
5. To develop an ontological model to capture different manufacturing operation and their sequencing knowledge.

1.4 Scope of the Research

This research focuses on the development of a framework and a solution model that can be applied across multiple domains of design and manufacturing. Specifically, the research has been concentrated towards enhancing the ability to share the knowledge residing within the different manufacturing domains to design engineers. Addressing the knowledge sharing issues pertaining to the domains of machining, assembly, welding and inspection together with their manufacturing process planning was the primary motive of this research. Real world scenarios from an aerospace discrete part manufacturing industry highlighted the knowledge sharing issues across the stated domains. Further it helped in developing the solution model and its validation. A case study exploited the proposed model's capability to highlight the implications of design change on the manufacturing domain and further provide recommendations to overcome any inadequacies. Although, the test cases and experimentation was based on an aerospace component but it is perceived to be applicable at a generic for any industry.

1.5 Research Approach

The adopted methodology for this research is built upon the previously defined objectives in Section 1.3. There are six main building blocks in the methodology that has been illustrated in Figure 2. The main aspect of the methodology is to develop a common verifiable semantic base in the form of an ontology that supports knowledge sharing across multiple domains. Thus, the development of the ontology is the element of emphasis in the methodology. Several ontology development methodologies are found in literature. In this research a manual ontology development methodology has been chosen as it provides a comprehensive and rigorous structure, which is required for a high level ontology (Blomqvist, 2008). Moreover, this approach better supports the identification and defining the crucial concepts with increasing specificity of the ontology. The literature survey revealed that the ontology development methodology proposed by (Blomqvist, 2008) and (McGuinness, 2002) was more relevant for the domain of manufacturing. Hence, their methodology with few other additions has been used in this research to develop the ontology. The additional steps involved are the formal declaration of the concepts along with the testing of their semantics.

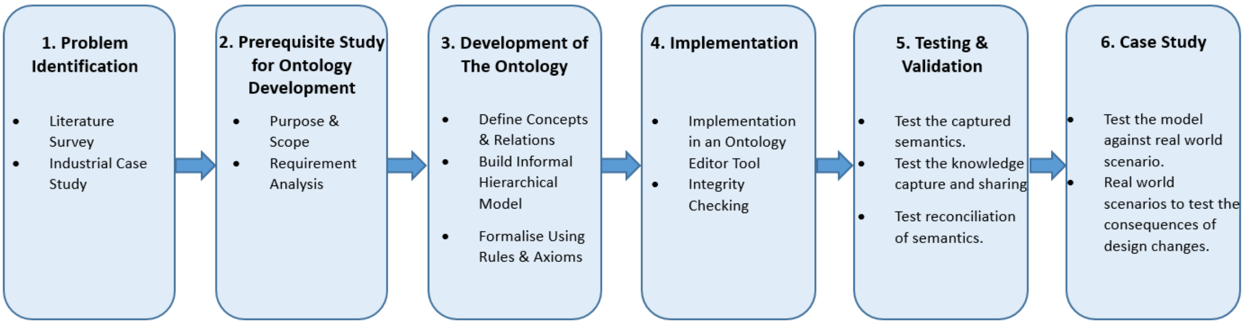


Figure 2Research Methodology

The above Figure 2 shows the entire methodology. The explanations of each building blocks of the methodology are as follows

1. Problem Identification: The first step of the methodology is to understand the current trends and the state of the art in manufacturing ontology. This is carried out by conducting an extensive literature survey and through an industrial case study. The study and the survey facilitate in ascertaining the gaps in the research and the issues in developing manufacturing ontologies.
2. Prerequisite Study for Ontology Development: The objective of the this step is
    a. To collate the purpose and scope of the ontology. This includes the users and the use cases of the ontology.
    b. Identify the fundamental issues related to knowledge sharing within the manufacturing domain from the industrial case study and literature review.
    c. Define the requirements for the proposed product lifecycle ontology unambiguously.
3. Development of the Ontology: The third and the most crucial step of the methodology is the actual development of the ontology. This includes several tasks to be completed such as
    a. Ascertaining the different categories or levels of concepts required within the ontology.
    b. Defining a set of concepts and relations based on the knowledge gained from the domain experts in the industry and the existing product lifecycle related ontologies.
    c. Creating an informal hierarchical model of the identified concepts and assign simple parent-child relationships. This also includes specifying the hierarchical category or level on which these concepts resides i.e the specialisation levels.
    d. Postulating the inter category relationships between the concepts.

   e. Finally, formalisation of the ontology by defining the constraints on the concepts through rules and axioms. The rigorous and unambiguous capture of semantics of the concepts is ensured by this.
4. Implementation: This step involves implementation of the ontology in an ontology development environment. In this research the Protégé ontology editor has been used as the development environment with Web Ontology Language Description Logic (OWL DL) as the formalisation language. More details about them in Chapter 4.
5. Testing and Validation: The implementation of the ontology is followed by asserting instances into the ontology to test and validate the hypothesis. It also involves analysis of the following aspects
   a. The ability of the proposed model to formally capture the semantics of the concepts.
   b. The realisation of the proposed framework to successfully capture the varying depths of meanings of the concepts.
   c. The capability of the proposed model to reconcile the semantics of divergent concepts through the consolidation of different standards.
6. Case Study: The final step of the methodology is to test the framework and the model in an industrial environment through a case study. It explores the models capability to share the knowledge across the different product lifecycle domains. The case study further involves real world scenarios. This investigates the model's competency to identify the consequences of product design changes on the other domains of its lifecycle and provide recommendations.

1.6 Novelties and Contributions of This Research

The research has culminated in the following contributions and novelties. These have been further elaborated in this thesis.

1. A novel ontological model in the form of Product Lifecycle Ontology (PLO) has been developed to capture the related knowledge from multiple domains of design, manufacturing, assembly with welding and inspection. Further, the knowledge captured from the domains of machining, welding and inspection domains has been shown to be shared with design.

2. A novel model for categorising the different joining and welding processes has been developed. This has further been used to consolidate the multiple welding standards to facilitate knowledge sharing.
3. An innovative approach to model and capture manufacturing operation and their sequencing knowledge has been showed. This is to aid better process planning activities.

1.7 Thesis Structure

After the inaugural first chapter, the thesis has been structured as follows: Chapter 2 provides an exhaustive literature review to identify the key research gaps which are required to be addressed. Based on this, the requirements are elaborated in Chapter 3. It also portrays a detailed overview of the PLO, description of the identified core concepts and one of the novel aspects of the research. Chapter 4 and 5 further elaborates the remaining two novel contributions of this research in details. The evaluation of proposed PLO is carried out in Chapter 6. Chapter 7 draws a conclusion to the research by reporting a discussion on the developed ontology and elaborating on the future scope of work.

2. Literature Review

2.1 Manufacturing Knowledge and Interoperability

Manufacturing is one of the key sectors which dictate the socio economic conditions of any nation. It plays a pivotal role towards the growth and prosperity of a country. In Europe, it is one of the top revenue and employment generators. Further, it has become a key part of countries' industrial strategy (Deloitte, 2017). Largely, manufacturing encompasses nearly 17% of the Gross National Product (GNP), 28% of total Gross Domestic Product (GDP) (Bank, 2019) and 23% of the total employment in the world (Bank, 2020). It's been reported in (Eurostat, 2019) that in 2018, that the manufacturing enterprises in EU had value added production of €5335 billion. A report published by (Rhodes., 2017) highlighted that manufacturing had accounted for 10% of United Kingdom (UK)'s economy in 2017, while it was 20% is Germany and 16% in Italy respectively. In 2018, manufacturing in UK had accounted for about 2.7 million jobs which was about 8% of total, had an economic output of £191 billion (10% of total), provided exports worth £275 billion (approximately 42% of UK exports) and had a product sales of £404.4 billion (Robinson, 2020) (Rhodes, 2020) .

The importance of manufacturing highlights that the associated knowledge is of paramount significance. A huge cost is incurred due to the lack in ability to share this knowledge and information. Furthermore, it drives inefficient product development. This introduces the concept of seamlessly exchanging information and knowledge across systems or interoperability. Interoperability is a derivative of the word "Interoperable" and commonly described in association with computer systems. There are various definitions for interoperability is found in literature (Ray, 2003) (Chen, 2004) (Chen, 2008) (Borgo, 2007). However, the most relevant definitions from the perspective of this research was proposed by (IEEE-Std-Computer-Dictionary, 1991) and (Chungoora, 2010). (IEEE-Std-Computer-Dictionary, 1991) defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged." (Chungoora, 2010) extended the definition from the perspective of design and manufacturing. He described interoperability as the "ability of knowledge base systems to seamlessly exchange design and manufacturing related information across these domains." The suitability of these definitions is because it encompasses the interaction across systems as well as their components. Further, a crucial aspect of interoperability is highlighted in this description, which is the ability of the computer systems to not only exchange information but also to understand them. Interoperability is further classified as follows (Briefing-paper, 2008) (Gürdür, 2018)

1. Syntactic Interoperability: The ability of systems to process a syntax string and recognise it as an identifier even if more than one such syntax occurs in the systems.
2. Semantic Interoperability: They signify the capability of the systems to identify if two concepts have the same meaning and further determine the relation if they are not.
3. Community Interoperability: It is the ability of systems to collaborate and communicate using identifiers whilst respecting any rights and restrictions on usage of data associated with those identifiers in the systems.

The concepts of knowledge sharing and interoperability play a pivotal role within the manufacturing industry as it utilises data and facts to define a set of activities for production implementation (Li., 2015) . There are several forms of manufacturing knowledge which transcends from product design, process planning, operations and service until disposal. Product data models such as STEP (ISO-10303-1, 1994) , part libraries eg. PLib (ISO-13584, 2001) , product knowledge models (CPM) (Fenves, 2006) are few such types of knowledge models related to manufacturing. From a business perspective, the capability to flawlessly share knowledge is a key contributing factor to improve the company's performance (Huang, 2010) . This is even more predominant for manufacturing industries as it dictates their competitiveness (Fathi, 2011).

2.2 Problems in Knowledge Sharing and Interoperability

Knowledge sharing across different manufacturing functions is an issue within the manufacturing community and is a difficult exercise. Its effect is more predominant within large manufacturing organisations having cross disciplinary teams, as they tend to work in silos (Saha, 2019). This is escalated when different departments have their own set of terminologies, perspectives and work context of the products (Szejka., 2014). Although the difficulty is pronounced in ICT based systems but their potential to support knowledge sharing outshines their drawbacks (Imran., 2015).

The capability to share manufacturing knowledge is a key factor that dictates the ability of the product to meet the customers' expectations, bring down the production cost and the time to market new product (Mesmer-Magnus, 2009) (Ayyaz, 2018) (Qadir, 2018). Moreover, the knowledge from the different product lifecycle domains is crucial for the designers. This is because although design accounts for only 5% of the total activity across the entire product lifecycle but its implication amounts to about 70% of the total product lifecycle costs (True, 2002). Furthermore, various design decisions are based on the knowledge from different product lifecycle domains at the disposal of the designers (Wang, 2008) . It has been reported that designers' use 30%-70% of their own personal knowledge while spending 70% of their time in searching updated knowledge from different product lifecycle domains. This was primarily due to the problems related to knowledge sharing (Lee, 2005). It has also been reported that about $31.5 billion is spent by the fortune 500 companies annually to solve problems related to knowledge sharing (Babcock, 2004). Furthermore, the problems with manufacturing knowledge sharing have been estimated to cost about $1 billion annually to the automotive sector of US (Brunnermeier, 2002) and $15.8 billion to their capital facilities (Gallaher, 2009) . The figure would substantially increase if other sectors and nations are considered. There are various reasons that lead to issues with knowledge sharing such as

1. Handling of incompatible data and information structures between different platforms. (Brunnermeier, 2002) (Cutting-Decelle, 2002) (Das, 2007) (Souri, 2017)
2. Incompatible syntaxes of the languages used in the software application systems. (Das, 2007) (Palmer, 2018)
3. Incoherent semantics of the concepts (definition of terms) used in the language of the systems. (Pouchard, 2000) (Fraga, 2020)
4. A lack of formalisation (computer interpretation) of the concepts. (Saha et al, 2017)

Among these, about 70% of costs have been reported to spend on the reconciliation of semantic inconsistencies (Bussler, 2005) . (Ray, 2003) (Szejka, 2017) analysed these inconsistencies and revealed the primary root cause. They owed it to be either because the same terms were being used to mean different things or different terms were used to mean the same thing resulting in semantic ambiguity. Thus, it can be understood that semantic issues is one of the prime barriers for manufacturing knowledge sharing and needs to be addressed. (Chen, 2004) suggested that the issues can be handled by using common or equivalent semantics. (Chungoora, 2010) proposed that rigorously defined semantics of the PLM system concepts could be a potential solution to achieve interoperability across product lifecycle domains. Formal semantics, which is also known as logical semantics, are generally defined as interpretation of meanings and expressions. It uses special logical systems aided by mathematical logics. The consequences of semantic interoperability on knowledge sharing were studied by many authors such as (Yang, 2006) (Lazenberger, 2008) (Ye, 2008) (Chungoora, 2013) (Usman, 2013) (Szejka, 2017) (Palmer, 2018). They revealed various potential methods to improve knowledge and information exchange. Some of these methods that are used to achieve semantic interoperability are discussed in the following section.

2.3 Methodologies for Achieving Interoperability

The process of knowledge sharing within and beyond the enterprises has existed for years. There are several ways by which this can be achieved. One of the most primitive and simplest approaches is from one individual to other via meetings, verbal discussions, e-mails etc. However, this method has several drawbacks as it relies on the availability of the individual, their capabilities and prone to errors. Furthermore, this approach is expensive and time consuming from the business perspective.

Model Driven Interoperability (MDI) methodology is based on a system development approach. It entails development of several integrated systems known as Model Driven Architecture (MDA). The Model Driven Software Development (MDSD) community introduced this approach and is currently recommended by the Object Management Group (OMG) (Bourey, 2006). It essentially supports the creation of machine readable models (Kleppe, 2003) which are transformed into domain specific models. This approach has been utilised by various researchers to address specific problems in different fields. (Cutting-Decelle, 2006) (Elvesæter, 2006) (Gnägi, 2006) (Didonet del Fabro, 2008) (Moalla, 2008) (Ducq, 2012) (OMG, 2012) (Bazoun, 2013). (Moalla, 2008) showcased that the MDI approach improved the quality of the product data for a vaccine supply chain. (Ducq, 2012) used this principles to model the transformations from business requirements to detailed specifications of multiple system components. Previous works highlighted the importance of this approach but it has its own drawbacks (Komatsoulis, 2008) (Usman, 2012) (Agostinho, 2016). Some of these are

1. They fall short in defining the domain concepts unambiguously.
2. They cannot be modified during runtime.
3. It does not support the reasoning and querying about the system structure, their components.

The aforementioned issues render this approach inefficient in achieving all the requirements for interoperability. Several frameworks and architectures have been developed for the purpose of interoperability. Some of these were developed from a technical perspective and the rest were from the business enterprise level. Some of the established architectures for the purpose of interoperability are

1. The Computer Integrated Manufacturing Open System Architecture (CIMOSA) (AMICE, 1993)
2. The Public Enterprise Reference Architecture (PERA) (Williams, 1994)
3. GRAI-GIM reference model (Chen, 1996)
4. Reference Model of Open Distributing Processing (RM-ODP) (ISO/IEC 10746-3, 1996)
5. The SUddEN approach architecture for interoperability (Weichhart., 2010).
6. Interoperable Architecture for operational processes (Gong., 2013).

ATHENA interoperability framework (AIF) was developed from the IDEAS interoperability framework as a part of the ATHENA project (Berre, 2007) from business perspectives. It builds on the IDEAS Interoperability Framework (IIF) and introduces a new element "*service*" in the framework. Along with this, the 'ICT systems' element is replaced with 'Information/Data'. This framework uses MDI and ontologies for interoperability across enterprises. It is known to support interoperability within and across organisations. (ISO/CEN-11354, 2008) was developed as a standard on frameworks for interoperability. It was built on from the AIF project. The standard proposes a multidimensional framework for enterprise interoperability where the interoperability approaches has been divided as:

1. <u>Integrated Approach</u>: As the name suggests, this is an approach that is dedicated to achieve interoperability through complete integration. Ideally, this would entail all the parties involved to be interconnected through a standard detailed structure to achieve a mutual goal. (Ozman, 2006). STEP, Parts Library and ebXML are some of the integrated approach based standards. However, this approach is not efficient for reconstructing existing systems. The common

standard structure makes it rigid. This makes interoperability between external organisations more expensive requiring a complete change of their adopted systems.

2. Federated Approach: In this approach, a dynamically evolving Meta structure is present between the participating partners that drive interoperability. Interoperability is achieved by providing the information at run time and any inconsistencies found are manually fixed. The Meta models can be used to map several entities and has a dynamically evolving structure. The extent of interoperability is maximum with this approach and suited for virtual enterprises. However, the dynamical nature of its structure reduces the practicality (Usman, 2012).
3. Unified Approach: This approach involves a common meta-level structure to which all the participants can map and build their own domain specific knowledge. The meta-level common structure is not executable but a commitment to this structure provides flexibility in modelling the domains and ensures interoperability. This approach is suitable for interoperability across different departments of manufacturing organisations (Usman, 2012). PSL (ISO 18629) is an example of this approach where it provides a common structure for interoperability across multiple process domains. Furthermore, this approach has been utilised in this research.

(Vujasinovic, 2007) worked on the semantic-mediation architecture and validated it in an industrial environment. They implemented their architecture in the ATHENA project. The platform of their implementation used semantic web tools and had eXtensible Markup Language (XML) and Resource Description Framework (RDF) capabilities. (Gupta, 2008) proposed a feature based framework based on the concept of "Domain Independent Form Feature (DIFF)". This was to support product model semantic interoperability. DIFF model acts as an interface between the source and the target systems. It has an underpinning of an ontology which provides a basis for representing features. However, the model was limited to the facial representation of features i.e. the differential referencing of feature shapes. The model did not delve in to design functions of features or the relationship between features and manufacturing processes. A Service Oriented Architecture (SOA) based framework was proposed by (Garcia-Dominguez., 2013) for interoperability between responsive manufacturing systems. In their work several agents and services were combined by mapping them to ISA-95 model concepts. Furthermore, rule based approach was utilised to determine the activity to be a service or an agent or both. There architecture sits well for enterprise level interoperability but it's usability for lower level systems needs to be explored. An attempt to improve interoperability between design and architectural model for construction industry was carried out by (Hu, 2016). They combined the foundation classes from the unified information model with various algorithms. This model acted as a centralised layer which standardized the entities, their attributes and relationships for conversion. Although the conversion algorithms are capable of handling semantic interoperability but it is very restricted to the domain of structural engineering.

A more commonly used method to achieve interoperability is through standards. A considerable effort has been made by various technical committees to utilise standards as a mechanism for interoperability. Standard for The Exchange of Product model data (STEP), Product Lifecycle Management (PLM), and Product Data Management (PDM) are some of the technical standards that has been created for product information and Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM) documentation. Perhaps the most relevant standard for product design and manufacture is the ISO10303 or commonly known as STEP (Pratt, 2001) . The essential aspects of STEP and its influence on interoperability have been studied previously by (Fowler, 1996) . Within the STEP standard, product data is represented from a neutral viewpoint across the product lifecycle in a standard computer understandable format (SCRA, 2006). This aspect of the standard is known to support interoperability across product lifecycle systems (Saaksvuori, 2008). The domain specific Application Protocols (AP) of STEP makes it more manageable and easy to implement (SCRA, 2006) . One of the most widely used AP is the AP203, it deals with assembly product related information (SCRA, 2006) . (SCRA, 2006) further demonstrated the utilisation of the APs which are based on "*machining features*". It is used in developing an integrated manufacturing architecture. Similarly, there are other standards that have attempted to achieve interoperability for product design and manufacturing (TC184/SC4, 2009), such as

1. Parts Library (PLIB)/ ISO 13584: This standard was developed to support interoperability between suppliers and users with respect to parts library data.
2. MANDATE (ISO 15531): Manufacturing management data interchange (MANDATE) was developed to represent production process data.
3. PSL (ISO 18629): Process Specification Language (PSL) was developed for representing the semantic definition of manufacturing process.

Similarly, there are various standards for welding that have been developed to support interoperability between welding and design domains. These standards also attempt to regularise welding processes as there are multiple categories based on material conditions and applications. The standards for welding have been discussed more elaborately in Section 5.

Although a variety of standards exists but their effectiveness for interoperability is questionable. Research has revealed that these standards have semantic consistency issues. The semantic inconsistency across manufacturing centric standards was investigated by (Chungooraa et al, 2013). (Young et al., 2007) had revealed that inconsistency of the word "*Process*" across the ISO 19439, ISO 18629 and ISO 10303 standards with regards to its informal semantics. A consensus from the user community to use a standard format for information representation would improve their utilisation. However, a lack of flexibility in the standards has deterred the user community to use them over the period. (Costa et al, 2007) reported the ambiguity of the definitions in ISO 10303 AP236 standard. Further, the subjective interpretation of the concepts within and across the standards brings more complexity (Ray, 2006). They further stated that even if the domain concepts are standardized, interoperability issues would still exist due to the differential understanding of meanings of the terms. (Michel, 2005), (Gunendran et al, 2007), (Young et al., 2007) have discussed the subjective interpretation of the concepts due to lack of rigorously defined semantics. The highly textual nature of the terms and definitions within the ISO standards makes them open to multiple human interpretations and sometimes misinterpretations (Michel, 2005) . This lack of formal definition of the concepts results in ambiguity making them inefficient and error-prone. Further, the resolutions of the issues are primarily dependent on "domain experts" who can agree on correct and consistent interpretation which is a deterrent towards interoperability across different ICT based systems (Chungooraa et al, 2013).

2.4 Knowledge Modelling in Design and Manufacturing

The inherent structure of the product lifecycle knowledge and information models is crucial for sematic interoperability. This is because the level of formality in the structure of the model dictates the semantic enrichment of the captured model. Thus, the modelling activity of the structure has a direct influence on their interoperability capabilities. Within the realm of PLM, traditionally the information is stored in two types of models

1. Product Models: These models store information related to specific product (Molina, 1995) (Anderl, 1997). (Balogun, 2004) further elaborated and defined it as "*a model representing a complex product from the top product level to the tolerance detail of every feature characteristic*". The product information that are generated, used and maintained across the processes of design, manufacture, delivery, maintenance and disposal are held in the product models (Lee, 2006).Further, the model allows them to be easily shared across the aforementioned domains. This nature of the model places them centrally within the product lifecycle domains (Young et al., 2007). The product models are comprised of following sub models.
    a. Structure-oriented: These contain information related to the structure of the product.
    b. Geometry-oriented: Geometry related information of the product is stored here.
    c. Feature-oriented: The information related to the several features of the products.
    d. Knowledge-oriented: These are models to capture the historical knowledge of the product.

    One of the most widely used product model is the Core Product Model (CPM) developed by (Fenves, 2006). It was aimed at providing a common ground and is capable of capturing the engineering context for product development. Further, it supports extensions to capture the different contexts of engineering for specific product views.
2. Manufacturing Models: The manufacturing models are the sources for the common manufacturing capability information and the constraint knowledge of manufacturing processes (Al-Ashaab, 1994) (Balogun, 2004) (Liu, 2004). Its foundations can be traced back to the work done by (Al-Ashaab, 1994). The information structure encompasses the relationships between the different components that dictate manufacturing capability. Similar to the product model, manufacturing model comprises of different sub models such as
    a. Manufacturing Resource Capability Model: It represents the information regarding the different elements of manufacturing resources and their contribution to the manufacturing process. (Zhao, 1999).
    b. Process Plan Model: The strategy information regarding the process plan for a manufacturing process is fabricated in this model (Feng, 2003).

c. Manufacturing Cost Model: The process of realistic estimation of production cost during design and manufacturing is based on the information provided by this model.

Based on these, (Feng, 2003) developed the “Manufacturing Object Model” to establish interoperability between design and process planning. They used the Unified Modelling Language (UML) Object-Oriented(OO) methodology to establish the backend information structure. This makes it an informal or lightweight model. Further, most of the previous work done on manufacturing models were based on OO approach and had led to creation of lightweight models. Although these models are developed in isolation but there has always been a requirement to integrate these models. The previous models were not integrated fully which is crucial for knowledge acquisition and decision support within product lifecycle development. The extent of semantic interoperability between the product and manufacturing models dictates the capability to capture and reuse the design and manufacturing knowledge. (Gunendran, 2006) in their work had established a framework to capture the different perspectives of design and manufacture. Additionally, they have mentioned the utilisation of different rules and equations to support the integration of multi perspective knowledge. However, the solutions that have been established to achieve this are based on UML. This results in an informal or lightweight model which is unsuitable for ICT system interoperability. Thus, a solution that would generate a formal or heavyweight model for ICT based system interoperability is still required to be addressed. A web based Knowledge Based System (KBS) was developed by (Reddy., 2018) for Computer Aided Design (CAD) and manufacturing systems. The system generates an intermediate CAD model from which Computer Numerical Control (CNC) codes are generated for manufacturing. Although their model utilised rule based approach using standards but it was mainly focused on design without delving into any semantics. Ontologies were used for proficient knowledge management as they provide terms to be accepted across enterprises (O'Leary., 2010) (Hinkelmann., 2016). Furthermore, ontological approaches for semantic inconsistencies and reconciliation were addressed by various researchers. This is discussed in next section.

2.5 Ontology Driven Interoperability

The accurate apprehension of the meanings concerning the information to be shared across heterogeneous systems is a driver for sematic interoperability (Szejka., 2017). A common semantic base in the form of an ontology is capable of achieving this by eradicating interoperability issues (Young, 2005) (Yang,

2006) and semantic mismatches (Chen, 2004) (Szejka., 2017). Ontologies have been used for proficient knowledge management as they provide terms to be accepted across enterprises (Huang & Diao, 2008). Within the realms of product lifecycle (design and manufacture), this methods have shown benefits by establishing a hierarchical structure that supports capture of commonly agreed knowledge (Chang, 2010). Ontological methods have further been shown to provide better illustration and justification of the complex relationships between different domain concepts (Chang, 2010) (Liao, 2016) (Palmer, 2016). Furthermore, they have been shown to be the central part of software systems and applications that support knowledge sharing (Benjamin, 2006).

The origins of "Ontology" can be traced back to Aristotle in his work on "Metaphysics". He described ontology as the science of "being *qua* being", which translates to be the study of the natural attributes of various things (Guarino, 2009). In other words it is the study of nature of beings (Oxford Dictionary, 2019) or the systematic account of existence (Ciocoiu, 2001). According to this definition and from the perspective of the philosophical community, an ontology focuses on the nature and structure of things independent of their actual existence (Staab, 2009) . However, this definition is not relatable for the computer science and hence ontology has a different perspective from their community. Ontologies have been referred as special kind of information objects which represents domain information for ICT systems (Chandrasekaran, 1999). From the perspective of ICT based systems, (Studer, 1998) explained ontology to be *"an explicit and formal specification of a conceptualisation."* This definition incorporates the application perspective of the ontology through three key words. It uses the word "explicit", signifying the exactness of the concepts and their interpretations within the ontology. Secondly, the term "formal" implies the machine readability of the ontology. And lastly, "shared conceptualization" that signifies the ontology has the capability to capture the agreed concepts for a particular domain. Similarly, there are various other definitions found in literature (Guarino, 1995) (Schreiber, 1995) (Heijst, 1996) (Uschold, 1996) (Guarino, 1997) (Roche, 2000) (Gruninger, 2001) (McGuinness, 2002) (Blomqvist, 2008) which were either an adoption of the previously mentioned definitions or newly defined.

The (ISO 18629-1, 2004) definition of ontology as *"a lexicon of the specialized terminology along with some specifications of the meanings of the terms in the lexicon"* is perhaps the most relevant description for this work. It portrays that ontology is capable of describing a set of concepts with axioms defining their meanings, thereby providing a basis for shared meaning (Young et al., 2007). It is quite commonly described as a multi-dimensional model for a particular domain of interest. The axioms constrain the meanings of the concepts to a particular domain. Therefore, the definition encompasses both the lightweight and the heavyweight aspects of ontology which are described in Section 2.5.2. It can be understood from the all the different definitions that essentially an ontology is comprised of a finite list of concepts and their relations (Antoniou, 2008). The 5 fundamental aspects of an ontology as described by (Liping, 2007) were

1. Concepts or Classes
2. Relations
3. Functions
4. Axioms
5. Instances

2.5.1 Classification of Ontologies

Several types of ontologies exist in literature based on different criteria's of classification. A comprehensive summary of all of the categories and the respective categories can be found in the work done by (Zhou, 2004) (Usman, 2012). However, the two main categorisation criteria which are relevant for this research work are

1. **Degree of Expressiveness**:
    a. Lightweight or Informal Ontology: These are simple taxonomies of concepts with basic relationships defined between them (Go´mez-Pe´rez, 2004) (Fernández-López, 2002). The relationships are generally in the form of hierarchical structures (parent-child) (Zhu, 2007). Lightweight ontologies are based on the assumption that the meanings of the terms of the concepts are understood readily. However, the weak or missing constraints over the concepts limit the correct semantic interpretation by ICT systems and deter interoperability (Dartigues, 2007) (Oberle, 2009). Some examples of lightweight ontologies are WordNet (Wordnet, 2010), ISO-STEP (ISO 10303), P-Lib (ISO-13584, 2001) etc.

b. Heavyweight or Formal Ontology: An underlying lightweight structure with an axiomatic layer on the top in the form of constraints results in a heavyweight or formal ontology. These models have rigorously defined semantics through rich logic. These provide the required restrictions over the meanings of the concepts (Go´mez-Pe´rez, 2004) (Borgo, 2007). The sematic rigour is brought about by capturing the meanings of the concepts and their mappings through formal mathematical logics (Zhu, 2007). Furthermore, the semantics of the concepts which are explicitly captured through axioms induces the capability to infer new knowledge (Zhu, 2007). The ontology is further entailed to support interoperability across multiple domains by interpreting the meanings of the different concepts (Dartigues, 2007) (Gunendran et al, 2007) (Chungoora, 2010).

2. **Level of Conceptualisation or Specificity:** This dictates the extent of specificity of the concepts prevailing within the ontology. These range from a very generic (upper or top) level to more specific (domain).

    a. Foundation Ontology: Foundation ontology is also known as upper or top level ontology. These are developed with the purpose to cover the semantics of everything (FinES-Cluster., 2011) (Sanchez-Alonso, 2006). It is capable of acting as a common semantic base for any domain as they are developed independently. However, the very generic and abstract nature of the concepts within this ontology renders them usable across a varied range of domains. Some of the widely acknowledged foundation ontologies are Descriptive Ontology for Linguistics and Cognitive Engineering (DOLCE), Basic Formal Ontology (BFO), Upper Level Ontology (ULO) from Highfleet, Open Knowledge Base Connectivity (OKBC) ontology, Suggested Upper Merged Ontology (SUMO), WordNet, Standard Upper Ontology (SUO). Concepts such as *Particular*, *Endurant*, *Perdurant* from DOLCE, *Abstract Entity*.

    b. Core Ontology: Core ontologies are placed somewhere in between foundation and domain ontologies to bridge the gap (Usman, 2011). They comprise of concepts which are neither as generic as foundation nor as specific as domain ontologies. (Gangemi, 2004) described that an intermediate set of concepts and relations between foundation and domain specific ontologies are classed as core concept ontology. These are also known as reference ontologies as coined by Nicola Guarino (Grenon, 2003). According to them, reference ontologies "*clarify the meanings of terms of a specific domain*". The semantics of these concepts are generic to be shared across multiple domains, as opposed to foundational concepts that cover the semantics of everything (Burgun, 2006) (Deshayes, 2007). It has been argued by (Deshayes, 2007) that core ontologies provide the formal semantics of the concepts and encourage their reusability along with shareability. (Leila, 2009) summarises that core or reference ontologies are broad enough to satisfy the needs of large domains, use axioms, supports shareability and can be derived from foundation ontologies. Some examples of core ontologies are Manufacturing Core Concepts Ontology (MCCO), Assembly Reference Ontology (ARO), Conceptual Reference Model (CRM), Condition Monitoring (CM) Core Ontology, Design For Manufacturing (DFM) Ontology etc.

c. Domain Ontology: These ontologies are developed for very specific domains with all the concepts related to that specific domain (Borgo, 2007). The concepts belonging to domain ontologies cover the specific semantics of the particular domain or application area being modelled (Musen, 1998) (Jean, 2006). For example, a product design and product manufacturing will have their own separate domain ontologies.

It has been identified that lightweight ontologies are incapable of rigorously defining the semantics of the concepts. This limits their capability to interpret the true semantics of the concepts from different heterogeneous systems, which is the basis for shared meanings of the concepts (Young et al., 2007). The shortcomings emphasises the need of a more rigorous approaches backed by mathematical models. This is provided by heavyweight or formal ontologies. This research work exploits the use of heavyweight ontological approaches to capture and share the manufacturing knowledge.

Also, it can be understood that based on the specificity of the concepts an ontology can be classified as foundation, core or reference and domain ontologies. And, the semantic specificity of the concepts increases from the foundational to the domain level. This research work is focused on sharing of product lifecycle knowledge and interoperability. Hence, it is crucial to understand the type of ontology which is suited for this purpose. Foundation ontology concepts have been found to be overly generic because the purpose of their conceptualisation was to cover a broad range of multiple domains (Borgo, 2007). Thus, the semantic base formed by foundation ontology becomes vastly generic to support interoperability across specific domains of product lifecycle. An ontology which is more focused than foundation ontologies is required. Domain ontologies have the capability to capture the specific semantics of a particular domain and hence are more specialised than foundation ontologies. However, multiple domain ontologies developed independently would not have any common basis for interoperability. Therefore, an ontology which is less specific than domain ontology and at the same time more specific than foundation ontology is required. This is where core or reference ontology comes into play. They have the potential to represent domain knowledge which can easily be shared and reused (Usman, 2012) (Brinkley, 2006). The primary aspects of core ontologies that were found to be in need of addressing were

a) To agree on a group of generic concepts within a certain community
b) To be able to share the semantics across different communities dynamically.
c) To be able align and map different ontologies
d) To be able to support multiple applications.
e) To create a template that is generic for a particular domain.

This research works is directed towards addressing points a, b, c and d. Although core or reference ontologies have been used in the field of medicine (Burgun, 2006) but very few have developed them within the product lifecycle world such as MCCO, ARO, DFM, ADACOR etc. These ontologies were specialised for certain manufacturing domains and developed independently without encompassing the others. These are elaborated in Section 2.6. This thesis aims at creating a core Product Lifecycle Ontology that encompasses all of the multiple domains as shown in Chapter 3.

2.5.2 Development of Ontologies

The process of development of ontologies encompasses three main elements as shown in the Figure 3.

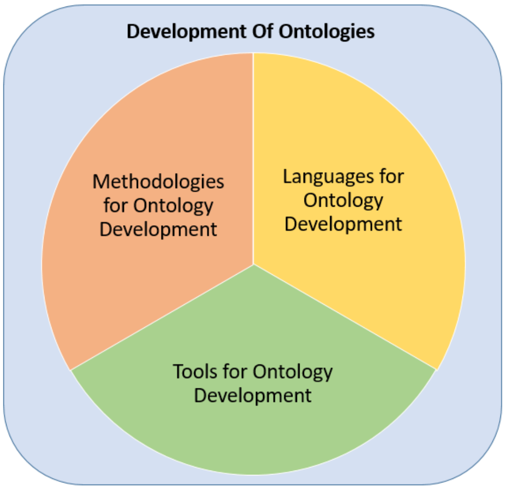


Figure 3Ontology Development

2.5.2.1 Methodologies for Ontology Development

The steps required to develop an ontology are described within the ontology development methodologies (Usman, 2012). Some of various ontology development methodologies which have been proposed are illustrated in the Table 1.

Table 1 Ontology Development Methodologies

No	**Methodology**	**Author**
1	Gruninger and Fox Methodology	(Gruninger, 1995)
2	METHONTOLOGY	(Fernandez-Lopez, 1997)
3	Blomqvist and Ohgren Methodology	(Blomqvist, 2008)
4	Noy and McGuinness Methodology	(McGuinness, 2002)
5	CommonKADS Methodology	(Schreiber, 2000)
6	Uschold and King Methodology	(Uschold, 1995)
7	IDEF5 Ontology Development Methodology	(IDEF5 Method Report, 1994)
8	KACTUS Methodology	(KACTUS, 1996)
9	SENSUS Methodology	(Swartout, 1997)
10	On-To-Knowledge Methodology	(Staab, 2001)
11	CyC Methodology	(Lenat, 1990)

12	Li's Methodology	(Li, 2007)

The methodologies those are more relevant for this thesis is discussed in Appendix A.

2.5.2.2 Languages for Ontology Development

The core structure of the ontology is represented by means of ontology development languages. These further provide the basis for knowledge capture and reasoning. Ontology development languages are categorised into 3 different fragments. The categories and a summary of the different languages within them are shown in Appendix B. The schematic languages represent the ontologies in a graphical format. The ontology markup languages on the other hand are based on eXtensible Mark-up Language (XML) and Description Logic (DL). Description Logic is a subset of First Order Logic (FOL) and provides the necessary representation formalism. This aids inference engines in decision making. The general ontology languages are mostly based on First Order Logic (FOL) and can be used to develop heavyweight ontologies. In this research work, the Unified Modelling Language (UML) and Web Ontology Language (OWL) has been used because of the following reasons

1. These models are easily understandable and usable for the population outside the Artificial Intelligence (AI) community
2. They have standardised graphical notations and
3. Various development tools are available (Go´mez-Pe´rez, 2004).
4. It provides more machine interpretability than XML, RDF and RDF(S) (Usman, 2012).
5. It can provide all the results with maximum expressivity and within a limited computational time.

These are further described in Appendix B.

2.5.2.3 Tools for Ontology Development

There are several ontology development tools that provide an environment to build ontology. Some of these tools are used to develop lightweight ontologies. While others provide an environment with inference engines, to load, instantiate and query the ontologies. The latter is used for developing heavyweight ontologies. (Corcho et al, 2003) (Frankovič, 2006) (Usman, 2012) provided a review of the ontology development tools. The table in Appendix B gives an overview of some tools which are relevant to this research. Protégé-OWL ontology editor has been used as the ontology development environment in this research. It is used to develop and deploy the ontology. Semantic Web Rule Language (SWRL) is used to define rules and axioms that provide the semantic rigor to the model. SWRL is an extension of OWL that permits complex rule definitions and advance reasoning over the concepts. The syntax followed in defining the rules are in the form of antecedent-consequent pairs. These rules help the system to interpret and infer the new knowledge. Furthermore, the Semantic Query-Enhanced Web Rule Language (SQWRL) is used to enquire the knowledge base for displaying the particular set of results. SQWRL is a query language for OWL which is based on SWRL. It is more concise, readable and semantically robust than other query languages such as SPARQL etc. (Connor, 2009).

2.6 Ontologies in Design and Manufacturing

Over the years, a considerable number of ontologies have been developed within the field of product lifecycle and manufacturing. Although, these ontologies have been developed to cater for different domains of product lifecycle but the prime focus has been areas of design and manufacturing planning. Some of these ontologies have been developed at an enterprise level while others for the purpose of knowledge sharing and interoperability at domain level. An overview of the relevant ontologies and their relevance or shortcomings with regards to this research is now elaborated.

One of the earliest ontologies relevant for this research was developed as a part of the TOVE (Gruninger, 1995) and Enterprise (Uschold, 1998) research projects. The primary objective of these projects was to develop a set of integrated ontologies (Common Sense Enterprise Model) to model commercial and public enterprises. The developed organisational ontologies were at an enterprise level with a taxonomical hierarchy. This makes the ontology to be very generic for modelling product lifecycle domains. However, the planning and scheduling axioms of the ontology was found to be beneficial to develop process planning concepts.

A manufacturing model was developed by (Molina, 1999) to capture the manufacturing capability knowledge, as a part of Model Oriented Simultaneous Engineering Systems (MOSES) project. Their ontology provided an understanding of some key concepts such as *Manufacturing Resources*, *Manufacturing Processes*, *Factory*, *Shop*, *Cell* and *Work Station*. However, the structure of the ontology was based on a lightweight ontological model rendering the system to be incapable of interpreting the meanings of the terms. Further, the model was not designed to share the information across multiple domains. (Yao, 1998) developed an aggregate product model for weld products. It had a lightweight structure but it provided some key concepts such as *Part*, *Weld Feature* and *Joint*, which were crucial for this research.

CIMOSA was developed for inter-enterprise interoperability and manufacturing enterprise modelling (Kosanke, 1999) . It was aimed towards providing a common understanding of business processes and enterprise knowledge. Some of the concepts that can be utilised for this research were *Manufacturing Enterprise*, *Products* and *Parts*. Similarly, the Factory Design Model (FDM) was developed by (Harding, 1999) to retrieve the information regarding an existing factory and the proficiency of a proposed factory. *Manufacture*, *Production*, *Production Plan*, *Material* and *Product* were some of the significant concepts that can be utilised from this ontology. However, both the above models were based on lightweight structures. Furthermore, these ontologies were focused on the enterprise level and not on different manufacturing domains.

The Process Specification Language (PSL) (Schlenoff, 2000) is one of the most revered heavyweight ontology for different processes. It was initially developed by National Institute of Standards and Technology (NIST) to capture the semantics of manufacturing processes but the generic nature of its concepts makes it suitable for other processes. PSL has been recognized as a standard for process semantics.

PSL and some of its key concepts have been found to be of key relevance in this research work. This was predominantly for capturing the operation sequencing knowledge as described in Chapter 5 . PSL has the capability to support interoperability across different process domains but it is inapt of defining secondary concepts and objects (Niles, 2001) (Schlenoff, 2000). Furthermore, PSL falls short in linking the processes with resources, tangible inputs and outputs (Young et al., 2007).

Manufacturing Systems Engineering (MSE) ontology was developed as a part of the MISSION project by (Jenny Harding 2003) (Lin, 2007). Based on a lightweight structure, MSE ontology was developed to act as a common meta-model for interoperability across manufacturing systems at an enterprise level. *Resource*, *Production Resource, Production Process*, *Product*, *Process* and *Parts* are some of the concepts which were found relevant for this research. This ontology was found to be very generic for multiple product lifecycle domains. Moreover, the lightweight structure of the model is barrier towards interoperability. (Feng, 2003) tried to integrate the design and process planning through a manufacturing process information model. In spite of being a lightweight structure modelled in UML, it provided a useful taxonomy of *Processes.* (Patil, 2005) developed the Product Semantic Representation Language (PSRL) ontology using DAML+OIL. PSRL was developed for product information interoperability using core concepts and relations from feature based modelling systems. Their ontology's drawback lies in its inability to capture the geometric feature information such as points and lines. *Feature* is a key concept from the PSRL ontology that has a significant use for this research work.

Manufacturing Semantics Ontology (MASON) is an upper ontology for the manufacturing domain. It was developed by (Leimagnan, 2006) as a common semantic ancestor for multiple domain specific ontologies. They used OWL as the formalisation language which enhanced its wider use. It had the capability to capture the individual operation level knowledge but not about part feature. (Semere, 2007) developed a machining ontology based on MASON. The MASON ontology was extended with machining concepts and also incorporated the aspects of features. However, both the above ontologies were incapable of sharing the knowledge to product design and encompassed only one specific domain.

ADAptive Holonic COntrol aRchitecture (ADACOR) is one of the prominent ontology for distributed manufacturing systems. ADACOR was developed by (Borgo, 2007) to support manufacturing planning and scheduling activities. Based on the DOLCE foundational ontology, it comprised of core concepts for production planning and scheduling. *Resource*, *Operation* etc are few of the concepts and their corresponding definitions which can be utilised in this research. ADACOR ontology is limited to only to scheduling and planning activities. Thus, its utilisation to share the knowledge to design remains to be explored. A lightweight Core Product Model (CPM) was developed by NIST for capturing the product model data (Fenves, 2006). Although the model comprised of various manufacturing core concepts but it was developed in isolation from other domains. It lacked the reuse of existing ontologies and focused primarily to capture product model knowledge. The ability to share the knowledge was also lacking.

However, CPM provides some key concepts such as *Feature, Form and Form Feature* which were crucial for this research.

(Chen, 2009) worked on the integration of different product lifecycle knowledge from different enterprises. Their work portrayed the development of a global ontology which can support the integration of different domain ontologies. This enabled knowledge sharing across different enterprise. Manufacturing of moulds was used to test the ontology. The model lacked concepts for other domains and did not capture operation sequencing knowledge. Moreover, the lightweight structure is incapable of interpreting the semantics. A novel approach for exchange of product information was proposed by (Tursi, 2009).The entire product was treated as an interoperable system for transforming Engineering Bill Of Materials (EBOM) to Manufacturing Bill Of Materials (MBOM). Their model was limited to only assembly domain but concepts like *Part, Product, Component,* and *PartVersion* has a good relevance to this research.

A methodology for information organisation that captures the best practise in manufacturing was proposed by (Gunendran, 2010). They had utilised libraries of best practice along with product and manufacturing models. A key element of their work was the relations defined between design and manufacturing features. However, the use of this approach to reveal the manufacturing consequences on the design phase was yet to be explored. The Semantic Manufacturing Interoperability Framework (SMIF) proposed by (Chungoora, 2010) is a relevant work for this research. A multilayer ontological framework was proposed for interoperability between design and manufacturing domains. However, their work was restricted to only simple ‘hole’ features and did not dwell into other domains. Another relevant work was the Design For Manufacturing (DFM) ontology by (Chang, 2010). They used their DFM ontology to analyse a particular design, for its manufacturability and further provide alternatives. Although, DFM had wide range of concepts but it was primarily developed for welding. Hence, it was restricted towards use in other domains. Additionally, their model they did not provide a structure that can semantically categorise the welding processes (Saha, 2019). (Usman, 2012) developed the Manufacturing Core Concepts (MCCO) ontology to share manufacturing knowledge to design. Similarly (Imran, 2013) developed the Assembly Reference Concept (ARO) to address the interoperability between the domains of design and assembly. However, all of these ontologies have been found to be constrained to specific domains and did not dwell into the sequences of manufacturing. Additionally, in ARO the concepts to model welding knowledge were missing.

An ontology based approach to model welding attribute defects from images was showcased by (Anouncia, 2010). Their work resulted in a domain ontology for welding defects with some key concepts relevant for this research. However, their work did not have any method to share this knowledge to design. (Hastilow, 2013) developed an ontology to address interoperability of manufacturing systems in different timeframes. He defined several levels within the ontology which was beneficial to understand the layered conceptualisation within a particular level of the ontology. As it was developed for manufacturing systems, it did not dwell into the different product lifecycle domains. An ontology based methodology and framework to develop platform independent Knowledge Based Engineering System was showcased by (Sanya, 2014). The model was focused on design parameters but was limited to simple shapes and geometry. Further, the models capability for design analysis from the perspective of manufacturability remains to be explored. (Julio Cesar Nardi, 2015) was one of the few authors who had tried to capture the "*Service*" knowledge and developed reference ontology for services. They developed OWL based heavyweight ontology but the share-ability with other domains is unknown.

(Solano, 2016) had extended the Product and Processes Development Resources Capability (PPDRC) ontology into an Manufacturing and Inspection Resource Capability (MIRC) ontology. It models the activities and capabilities of manufacturing resources, which supports machining and inspection planning activities. The taxonomy of resources provided a good basis for understanding the relation between the resources and inspection planning concepts. However, the method to share this knowledge was missing. (Dinar M, 2016) developed a Design For Additive Manufacturing Ontology (DFAM). This ontology was developed to provide flexibility to designers in order to create complex geometries which are limited in conventional manufacturing. Despite being developed solely for Additive Manufacturing, various concepts in DFAM such as *Design Feature, Manufacturing Feature* and their relations can be utilised for this research work. (Yang, 2016) developed a meta-model for manufacturing process information, while (Manupati., 2016) proposed a mobile-agent based integration approach using an ontology for process planning and scheduling. Both of these works did not delve into the granular details of different manufacturing operations and it sequencing. (Szejka., 2017) developed a semantic reconciliation view to support interoperability between design and manufacturing using ontologies. A combination of ontology intersection, adjustment context and semantic alignment was utilised to support the semantic relationships.(Szejka., 2017) further showcased the use of reference ontologies for semantic interoperability during the process of product development. Their approach was based on semantic mapping with focus on tooling and injection moulding. However, the utilisation of the approach on consequences of design changes on manufacturing remains to be explored.

A foundational ontology to aid modelling manufacturing systems was proposed by (Zaletelj, 2018). Some of the key highlights of their model were (1) the formalisation of relationship between model and system, (2) capability to model system behaviour through time and (3) a metamodel for manufacturing systems. The layered approach using model, metamodel and metametamodel provided some key understandings for the developing the ontology in this research. (Palmer, 2018) developed the FLEXINET reference ontology to capture the logical relationships between several manufacturing concepts. The aim of this ontology was to aid businesses in strategic and tactical decision making. Thus, it was more focused on communication between different manufacturing systems with little focus on implications of design changes on the other domains of the product lifecycle. Nevertheless, an unique element of their ontology was the usage of the concept *Role* to capture the different perspectives of various concepts which has been exploited in this research work. An ontology based methodology was proposed by (Liang., 2018) to model the process planning activities for additive manufacturing but did not divulge into the different operation types. (Ali, 2019) developed a product lifecycle ontology but focused on the additive manufacturing processes. Although it was product lifecycle ontology but the concepts were not generic to encompass other manufacturing process. Similarly, (Saha, 2019) developed the Core Domain Ontology for Joining Processes (CDOJP) to capture knowledge regarding joining processes. It provided various key concepts for welding knowledge modelling but was limited to the specific domain.

A Neo4j based approach using ontologies was proposed by (Zhu., 2019) to address the process scheduling and planning problems in a flexible manufacturing environment. A Semantically Integrated Manufacturing Planning Model (SIMP) was developed by (Šormaz., 2019) to model the constraints of manufacturing process planning i.e. variety, time and aggregation. (Sarkar., 2019) proposed an extension of the Basic Formal Ontology (BFO) to model the process level capabilities of manufacturing resources.

(Souri., 2019) proposed a mechanism for integrating manufacturing knowledge with the design process. Although the authors had proposed a collaborative solution for propagate workflows between different departments but the mechanism was from a high level perspective. An ontological model was utilised for smart planning, dynamic monitoring and value stream mapping using Neo4j by (Zhuoyu., 2020). (He., 2020) proposed an ontology based method to efficiently reuse the remanufacturing knowledge and model its process planning. (Šormaz., 2020) proposed a Semantically Integrated Manufacturing Planning Model (SIMPM) to model the fundamental constraints of manufacturing process planning based on variety, time and aggregation. Although their model catered for different machines, materials, features and processes but the model did not delve into the types of operations. The detailed sequences of those operations and classifications from their model remain to be explored.

Most of these previous works are either at a holistic level providing an overview of the process information, or they are narrowly restricted to a specific domain. A general lacking of a model that captures the intricate details of generic forms of operations in manufacturing enterprises has been perceived. Additionally, the absence of capturing operation sequencing knowledge is a barrier towards supporting effective process planning.

2.7 Features Based Engineering and Ontologies

The review of previous research work has highlighted that most of the knowledge models are developed independently for specific domains. Primarily the information are either held in product models for Computer Aided Design (CAD) systems or in manufacturing models for Computer Aided Process Planning (CAPP) or Knowledge Based Engineering (KBE) systems (Al-Ashaab et al., 2003) (Balogun, 2004) (Liu, 2004) (Sudarsan, 2005) (Srinivasan, 2011). The individualistic development of these models results in a gap which is essential to be overcome. Feature based engineering method is commonly utilised to bridge this gap (Shah, 1995) (Otto, 2001) (Dartigues, 2007) (Abdul-Ghafour, 2011), as this results in an integrated representation of the engineering product data (Sanfilippo, 2016).

A feature can be understood as an information unit described by an aggregation of properties and represents a region of interest within a product (Brunetti, 2000). Feature based engineering has two main approaches. These are

- Feature Recognition and Extraction: This is an algorithmic approach which uses mathematical expressions to identify the features followed by their extraction. However, the effectiveness of the algorithms to identify the features is uncertain (Martino, 1998). Furthermore, in this approach the manufacturability of the features can only be extracted after the part has been completely designed (Usman, 2012).
- Design by Features: In this approach, the features are extracted from a library of manufacturing features to model a product. Thus, the manufacturing information is available to the designer as the product is modelled from those predefined feature library embedded with the manufacturing information. However, the feature is based on manufacturing context. This is a drawback towards the designer’s modelling accuracy and the features functionality. Moreover, a predefined library of features restricts the flexibility of the designers.

Various research has been directed towards feature based design and manufacturing (Wang, 1993) (Salomons, 1993) (Chen, 1997) (Gunendran, 2008). Furthermore, the utilisation of features as a link to integrate design and manufacturing has been portrayed in the works of several researchers such as (Young, 1993) (Gu, 1994) (Aifaoui, 2006) (Ma, 2007), PERA approach by (PERA., 1969), Opitz classification by (Opitz, 1970), Brisch System by (Gallaghar, 1986), MICLASS approach by (Houtzeel, 1975), DCLASS by (Love, 1985), FORCOD by (Jung, 1991).

Similar approaches have been explored within the ontological world for interoperability (Gunendran, 2008) (Dartigues, 2007) (Abdul-Ghafour, 2011). These approaches to share manufacturing knowledge to design have been more inclined towards design by feature approach. As mentioned earlier, this is a context dependent approach. Thus, predefined design features compromises manufacturing and vice-versa. (Usman, 2011), (Jagenberg, 2009) proposed the use of “standard features” to overcome this issues, as they are perceived to be standard across design and manufacturing. The forms of these standard features, manufacturing methods and the functionality are already agreed by designers and manufacturing engineers. However, the standard features have been found to be limited to few features as multiple product lifecycle domains have different perceptions and interest (Usman, 2011).

Several authors have used mapping approaches to achieve interoperability between design and manufacturing. A feature based ontological approach using commonly understood feature ontology was proposed by (Dartigues, 2007). Their ontology was developed to exchange information between CAD and CAPP systems. The feature ontology was heavily based on the CPM and STEP standard. Perhaps, the most relevant work within this area was that of (Chungoora, 2010) (Gunendran, 2010) (Chungoora, 2011) (Usman, 2011), (Imran, 2013). (Gunendran, 2010) showcased a model based on part families and features that had the capability to highlight the consequences of design changes on manufacturing. Their feature related concepts and relations had been further adapted and extended by (Usman, 2011) to develop MCCO and by (Imran, 2013) to develop ARO. These have been further explored, modified and extended in this research work as described in Section 4.

Features have varied perspective from different domains of the product lifecycle as mentioned previously. For example, a design feature is based on a designer's perspective, which is its functionality. Whereas, the same feature is treated as a manufacturing feature based on a manufacturing engineer's perspective, reliant on its method of manufacture (Usman, 2013). There is a differential representation of features across multiple product lifecycle domains. But the relevant research works mentioned before have tried to define a common terminology to relate the domains. This aspect is further explored in this research to develop the Product Lifecycle Ontology (PLO).

2.8 Summary and Research Gaps

This chapter provided an extensive review of the existing researches within the field of knowledge sharing across product lifecycle domains. The extensive review assisted in ascertaining the various knowledge sharing issues of ICT based systems across design and manufacturing domains.

The review highlighted the inability of ICT based systems to be effectively share knowledge and achieve interoperability. This was found to be a common problem across different organisations, costing them a fortune. Thus, an efficient measure is required to address these issues. Several approaches and standards to achieve interoperability issues have been discussed. However, these methods were found to be inept as they were limited in terms of rigorously defining the concepts.

Ontology based approaches that utilises heavyweight ontologies have the potential to overcome the semantic issues and achieve interoperability. However, the level of their expressivity dictates the extent to which they achieve this feat. Core or reference ontologies, which bridge the foundation and domain ontologies, were identified to be best suited for this. This is further explored as a solution for this research work. Various methodologies for developing ontologies have been explored. This revealed that the method proposed by (McGuinness, 2002) and (Blomqvist, 2008) with some additional steps is best suited for this research. The additional steps involved were the formal declaration of the concepts along with testing their semantics (Saha, 2019). Subsequently, the various languages and tools required for ontology development was elaborated. This highlighted the benefits of OWL and Protégé as an ontology development language and tool respectively. Thus, they were preferred for this research.

The section progressed further to provide a comprehensive review of existing ontologies that were developed for Design and Manufacturing domains. It highlighted the following primary gaps that are addressed in this research work,

1. More research is required within the field of ontology based methods to support interoperability across design and manufacturing domains.
2. Most of the developed ontologies have been found to be either very generic or too specific for a particular domain. Hence, there is a need to explore core or reference ontologies to address the knowledge sharing issues from multiple domains.
3. Multiple viewpoints of features from the perspective of design and the different domains of manufacturing (machining, welding, inspection) is required to be studied to address the knowledge sharing issues.
4. Although research has been done towards developing ontological models for assembly processes, but an investigation to understand the semantic inconsistencies and their implications on interoperability for welding processes was absent.
5. There is a general lacking of a model that can capture the intricate details of generic forms of manufacturing operations that exists in manufacturing enterprises. Similarly, there is an absence of a model that can capture the operation sequencing knowledge with high granularity.

The following sections address the above mentioned research gaps.

3. Product Lifecycle Ontology: A Core Ontology to Share Design and Manufacturing Knowledge

This chapter provides a brief overview of the issues pertaining to knowledge sharing between design, machining, welding and inspection domains. This is followed by identifying the requirements for the core Product Lifecycle Ontology to address these issues. The solution framework and model in the form of a PLO is introduced and the following novel contributions are discussed.

- PLO comprises of an intermediate set of core concepts from different domains of manufacturing and design. These are reusable concepts that ensure capture and sharing of the knowledge from design, machining, welding and inspection domains. A single central model comprising of such diverse set of concepts from multiple domains is one of its novelty.
- Several new specialisation levels have been proposed within the core and domain level to capture the knowledge with higher granularity.
- New categorisation of the concepts has been proposed which eliminates semantic mismatches. Further, it ensures that PLO can act as a semantic base for development of application specific ontologies and provide a route for knowledge sharing across those.

The chapter is organised as: Section 3.1 discusses the issues and requirements for knowledge sharing across the machining, assembly, welding and inspection domains. Section 3.2 introduces the PLO framework and discusses its novel aspects. The modelling process is discussed, method to represent and share the knowledge and the specialisation levels are discussed further. Section 3.3 elaborates the PLO concepts, the relationships between these concepts along with their formalisation. Section 3.4 explains the role of PLO to provide the route to share knowledge. The model is experimentally validated in Section 3.5. And, finally Section 3.6 summarises this chapter.

3.1 Issues and Requirements for Knowledge Systems to Share Design, Machining, Welding and Inspection Knowledge

Design and manufacturing are perhaps the most significant domains with respect to cost drivers for an enterprise. Additionally, manufacturing itself encompasses domains such as machining, assembly with welding and inspection. Thus, a collaborative approach underpinned by seamless knowledge sharing amongst these domains is absolutely essential for efficient product development. The emergence of ICT systems has resulted in development of various knowledge based systems to store and reuse this information. The notion of using ontology based knowledge systems have been researched extensively for this purpose, as elaborated in Section 2.6. However, these systems have been found to be incapable of addressing the requirements of modern manufacturing systems, as they are developed to operate in isolation. The issue paramount's when the number of domains being addressed increases. This is primarily because most of these ontologies were either not developed to address the requirements of knowledge sharing (Chungoora, 2010), or were constrained to a particular domain (Saha et al, 2017). This was highlighted in Section 2.6. Moreover, the ontologies were found to have varying degree of expressivity which limits their synergy with other ontologies (Ray, 2004). Therefore, a semantically enriched ontology based knowledge sharing system for the aforementioned domains of the product lifecycle is still wanting.

The prime barrier towards knowledge sharing across these ontological systems is their inability to reach an agreement on the semantics of the gathered knowledge (Musen, 1992). Multiple domains imply different context and semantics of the knowledge. This is misconstrued when they are required to be shared. To ensure knowledge sharing, it is crucial that the semantics of the knowledge is preserved. The concepts which are used to capture the knowledge from the multiple domains have different implications in other domains as shown in Figure 4. For example, the concepts such as *Product*, *Feature*, and *Tolerance* have a functional attribute from a design perspective but for other domains its attribute is more aligned towards their respective processes and resources.

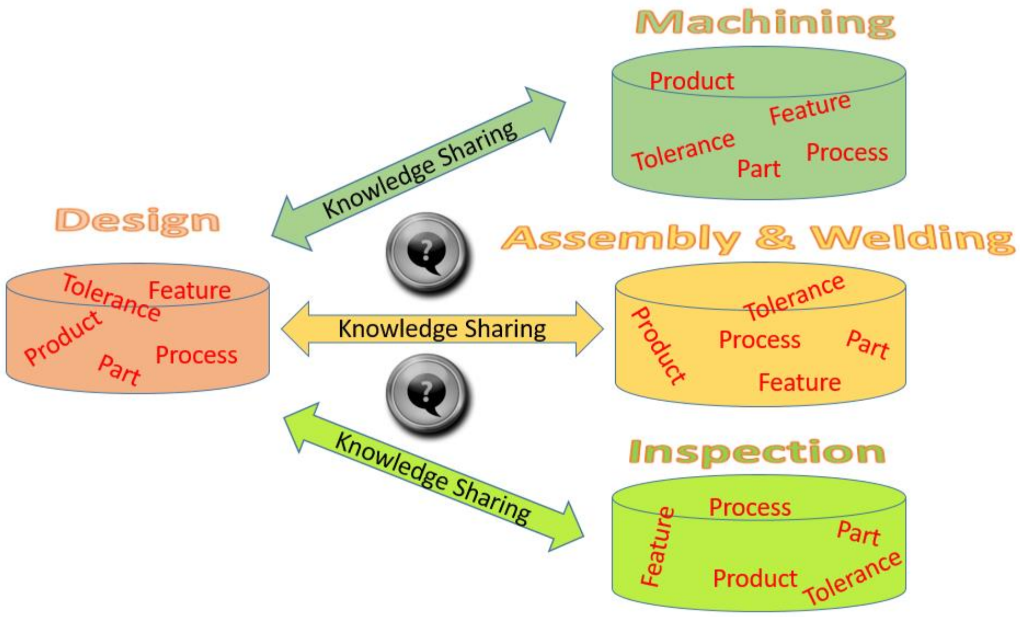


Figure 4Problems with Knowledge Sharing from Multiple Domains

The variable context of concept semantics is further elaborated through the following example from the industrial case study in Figure 5. The figure shows a portion of an aero engine static compressor component, an Outer Guide Vane (OGV) Assembly. Here the outer guide vane (OGV) is welded on to an inner ring. The other end of the vane is assembled onto the fan case. The figure shows the different views and perspectives of the same part from the four different domains. It shows the mismatches that occur across all these domains. Two types of mismatches can be observed

1. Mismatch arising from referring the same forms differentially. For example, the designers term the flat top surface as the ‘Abutment Face’ while the machinists designate it as ‘Vane Attachment Area’, the assembly engineers state it as ‘Mating Face’ and the inspectors term them as ‘Face Height.
2. The differential semantics of the terms across these domains. This can be elaborated from the different perspectives from domains as illustrated in Figure 5. For example,
    a. The design engineers are interested in the functional attributes. Such as, what function the part has to perform during service? What temperature the part has to operate at? What are its weight restrictions? What material will it be made of?

b. The manufacturing engineers on the other hand are concerned with the process they need to follow in order to realise this part. The machining engineers need to know type of machining processes would be required for this part. Also, the cutting tools, machine tools and their capability to produce these parts is their concern. The assembly engineers are more interested about the process required to join the parts. The FIT required by the parts, the assembly tools and resources necessary to carry out the operation is also there area of interest. Lastly, the measurement engineer's deal with the inspection processes required to prove if the part is confirming. The different inspection tools, machines and their capabilities to accurately measure to the design tolerances are the notion of importance to the inspectors.

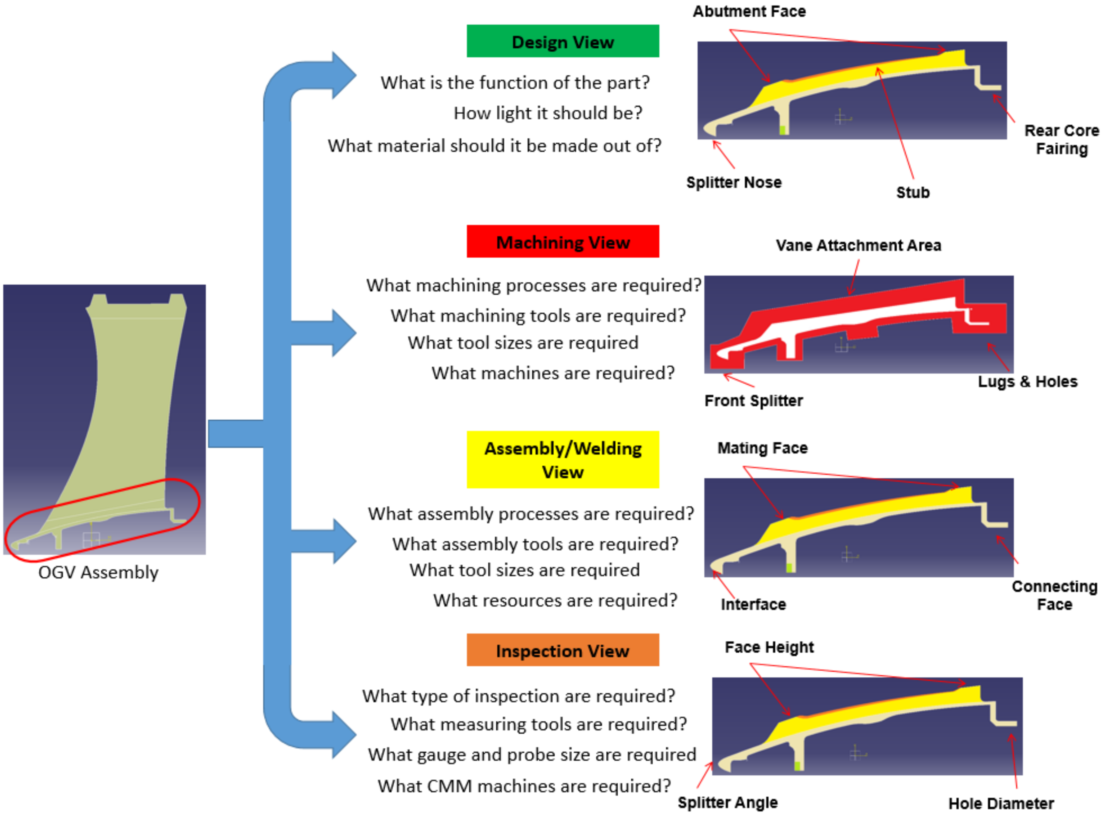


Figure 5Different Perspectives of the OGV Assembly from different domains

The above description portrays that the semantics of the concepts are primarily dictated by their respective domains. As a result, there can be different associated concepts linked with the primary concepts. This leads to different domain specific data structures. For example, *Mating Configuration* is an assembly domain specific concept associated to *Assembly Process*, while *Aero Feature* is for design domain. Therefore, the contexts of the semantics are required to be captured to overcome any ambiguities. This highlights that the ontological system is required to capture not only the semantics but also the context of the knowledge.

Although standards can be utilised to overcome the semantic conflicts, but this would require all the participating system to adopt the same standard. Moreover, the standards themselves have inconsistent and incoherent semantics, which requires 'domain experts' to resolve. These have been discussed in detail in Section 4. The semantic conflict issues can be resolved by ontological models. From the discussion in Section 2.4 it was understood that domain ontologies are developed for specific domains. These concepts capture the semantic perspectives of the specific domain. Contrarily, foundation ontologies capture a very generic perspective of the concepts. These generic semantics of the concepts are agreed by different domains. However, the captured semantics are overly generic from the perspective of knowledge sharing across the domains. For example, *Event* is a foundation concept that can encompass various things such as *Process, Festival, and Concert* etc. More specifically for product lifecycle if *Process* was classified as a foundation concept then this could refer *Manufacturing Process¸ Design Process and Transportation Process* etc. Thus, a large number of concepts with different perspectives from multiple domains would be inferred to be the same. This would instil semantic ambiguity which prevents knowledge sharing. Therefore, a core set of concepts capable of capturing intermediate semantics is required to (1) capture the different perspectives, (2) overcome semantic mismatches and inconsistencies and (3) act as a semantic base to ensure sharing of knowledge. The commonality of these core set of intermediate concepts ensures that the semantic consistency is preserved while extending them into domain specific concepts. This allows the systems to ascertain the concepts with similar or discrepant semantics across different domains. Therefore, the following framework as shown in Figure 6 has been proposed. The model is further elaborated in Section 3.2.

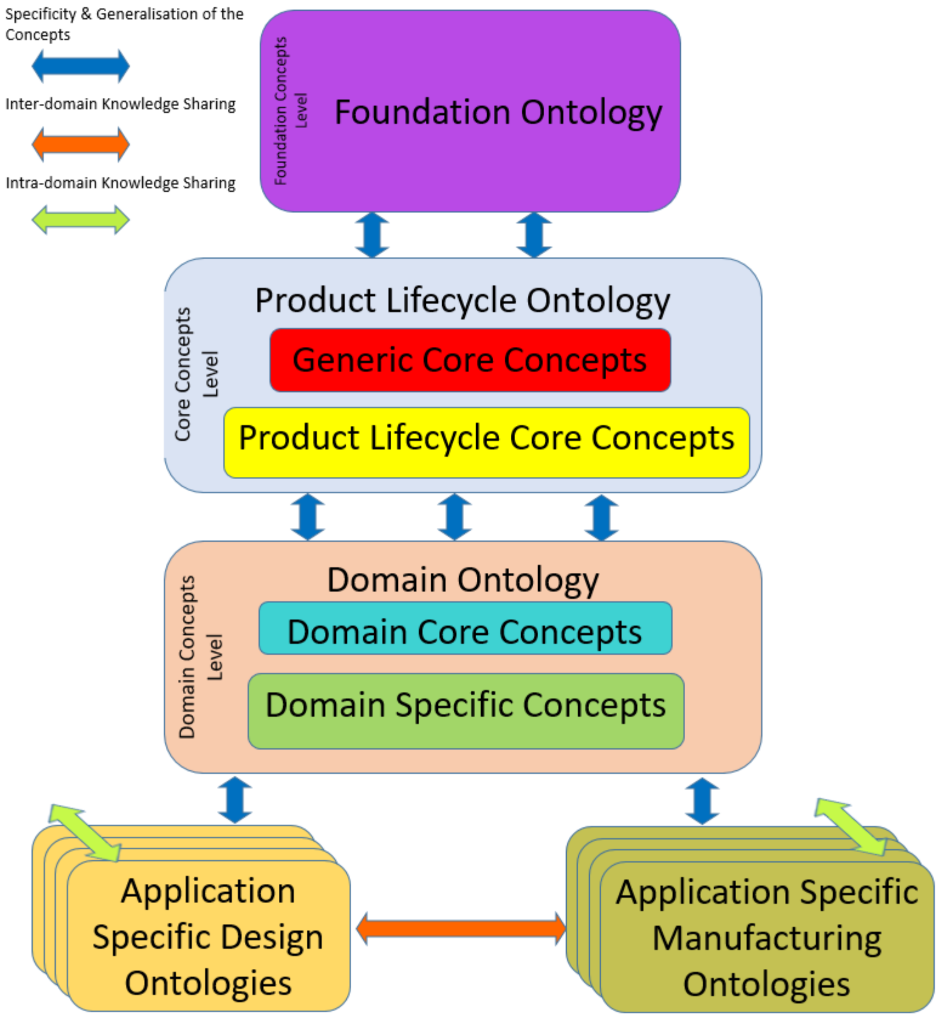


Figure 6PLO framework to share knowledge

After establishing the need for reference ontology with a core set of concepts, it is important to identify the level of expressivity or semantic rigor required to define those concepts. From the discussion in Section 2.4, it was understood that lightweight formalisation provides simple taxonomies with loosely defined semantics. Such formalisation results in semantic ambiguities for the concept definitions, as they are open to differential interpretation and misinterpretation. This is a barrier towards knowledge sharing. Thus, a more rigorously defined semantics through formal logics are required. This would enable the systems to identify the differences or similarities between the concepts from multiple domains (Oberle, 2007). A heavyweight ontology where concepts are defined with formal logics has the capability to appropriately interpret the meanings of the concepts. However, to attain seamless sharing of knowledge, it is imperative that the ontology is developed collaboratively.

Therefore, based on the highlighted issues the following requirements for an efficient system to support knowledge sharing can be summarised

a) There is a need to capture the different perspectives of the concepts. This implies that the semantics of the concepts along with their context and relations are required to be captured to ensure sharing knowledge across multiple product lifecycle domains.

b) A set of reusable and overlapping concepts are required to be identified from the domains of design, machining, welding and inspection. This must have the capability to provide a route for sharing knowledge.

c) An appropriate formalisation language is required to capture the semantics of concepts from multiple product lifecycle domains. It must have the capability that enables the system to understand the meanings of the concepts computationally. Further, it should be able to identify the similarities and differences between concepts from all these domains.

3.2 Product Lifecycle Ontology: A Framework to Share Knowledge across Multiple Design and Manufacturing Domains

This section introduces the proposed Product Lifecycle Ontology (PLO) to overcome the issues described earlier. Based on the requirements, core (reference) ontology was found to be more suitable. However, after delving more into the concepts required to capture the product lifecycle knowledge it was understood a further specialisation within the reference layer was required. This was primarily to facilitate the capture of knowledge with more granularities. Similarly, a certain level of specialisation was required at the domain layer as well. Therefore, PLO and the domain ontologies comprised of different layers as shown in Figure 6. At the very top is the foundation layer, which specialises into the core concepts level. The core concept layer has several levels within and finally specialises into multi-levelled domain layer. The application specific ontologies are further developed from this. The philosophy behind this approach is that since the application ontologies are developed or adopted from the PLO it would still retain the semantic integrity. Consequently it would serve as the common base providing a route to share knowledge. Moreover, PLO was developed to act as reference ontology for inter-domain and also for intra-domain knowledge sharing. Therefore, few concepts from other domains have been made part of the PLO. The foundation layer comprises of basic concepts from the Protégé and DOLCE’s foundation ontology. A more elaborate description of the different layers and their concepts are explained in the next section. The purpose of PLO was to eradicate issues pertaining to knowledge sharing across multiple product lifecycle domains. Thus, the novel aspects of this model and framework are

1. The proposed framework identifies an intermediate set of core concepts from the different product lifecycle domains. These are reusable, which act as reference concepts supporting representation of knowledge and it's sharing.
2. This framework portrays the need to have specialisation of the product lifecycle concepts both at the core and the domain level to capture the knowledge with higher granularity.
3. Within the proposed model, the proper categorisation of concepts through the specialisation levels supports the elimination of semantic mismatches. It further reconciles the semantic inconsistencies and incoherencies across multiple standards. This is elaborated in Section 4.
4. The proposed PLO acts as a semantic base for development of application specific ontologies and act as a route for knowledge sharing across those. This is explained in Section 3.2.2.

The modelling process and various concepts of PLO are described in the following sections.

3.2.1 Specialization of Concepts

The notion of various specialisation levels within PLO has already been mentioned and illustrated in Figure 6. The concepts are specialised from a very generic to highly specific forms. Previously, specialisation levels have been proposed by (Usman, 2012) and (Imran, 2013). But the proposed levels were for single piece and assembly parts respectively. Hence, they were not applicable directly for PLO which encompasses a broader realm. The need for modification and new levels of specialisation within the PLO is illustrated through Figure 7 and the following explanation. Figure 7 shows the varying depth of meanings within the core concept level. It can be seen that from the semantic neutral viewpoint of the concepts at the core concept level, there is a challenge for capturing the detailed knowledge with high granularity. For example, the concepts *Process* and *Manufacturing Process* both reside at the core concept level but they have varying depth of semantics as one has more specific realm than the other. Similarly there are other concepts within the same level which has different specificity. This can potentially lead to improper knowledge capture at higher granularity. Hence, it is crucial to capture the variations for ensuring seamless and consistent knowledge sharing. Multiple levels of specialisation are therefore proposed to capture this variation of meanings through the evolution of the concepts from the very generic to the more specific domains.

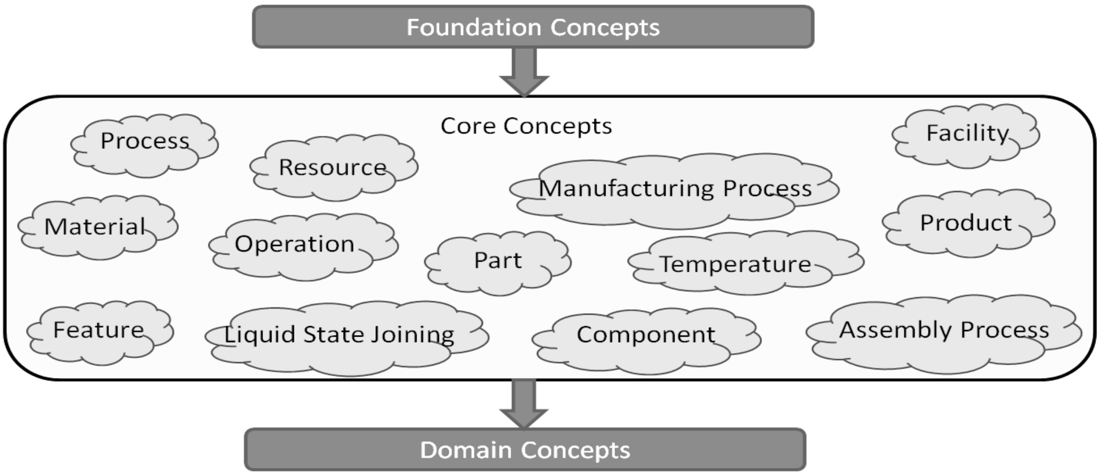


Figure 7 A challenge to represent varying depths of meaning within the core concepts level

Figure 8 illustrates such multiple levels of representing the *Process* concept from its foundation to the domain level, which is explained below. The generic semantics of the concept *Event* has a wide spectrum of application and acts as the foundation for the *Process* concept specialisation. In this way the foundation level is specialised into the Core Concept level, which has Generic Core Concepts and Product Lifecycle Core Concepts as its sub layers. *Process* forms part of the generic core concepts as its semantics is prevalent across different processing and manufacturing domains. It is further specialised into *Manufacturing Process*, which is a product lifecycle core concept for addressing the semantics of multiple product lifecycle domains. The Core Concept level is specialised into Domain Concept level. Similar to the previous level, this has Domain Core Concepts as one of its sub layer for concepts generic to a particular domain. Domain Specific Concepts is it's another sub layer for concepts that are highly specific and semantically constrained to only one particular domain. The *Manufacturing Process* evolves through these layers as *Assembly Process* and *Welding* respectively. The concepts have further provisions to develop application specific ontologies. In summary, the above concepts are defined below:

a) Foundation Concepts – Concepts that are vastly generic for any application, e.g., *Object*

b) Core Concepts

i. Generic Core Concepts – Concepts which are generic, irrespective of the type of applicable industry, e.g., *Process* is a concept that has its utilisation in the mechanical, manufacturing, software industries, etc.

ii. Product Lifecycle Core Concepts – Concepts which are generic across multiple Product Lifecycle domains, e.g., *Material* which has its applicability across the entire product lifecycle

c) Domain Concepts

i. Domain Core Concepts – Concept which are generic for a particular domain. E.g. *Mating Configuration* is generic for the entire domain of joining

ii. Domain Specific Concept Level – Concepts which are constrained to a particular domain. E.g. *Welding* which is one specific joining process.

An important aspect of the concept specialisation that needs to be highlighted is that all the concepts do not necessarily follow all the layers of specialisation. It means that some of core concepts may not have their parent class from the immediate generic layer. For example, *Product* is subclass of *Physical Object*. This means it is a direct descendent of the Foundation level class and bypasses the Generic Core level, the immediate generic level.

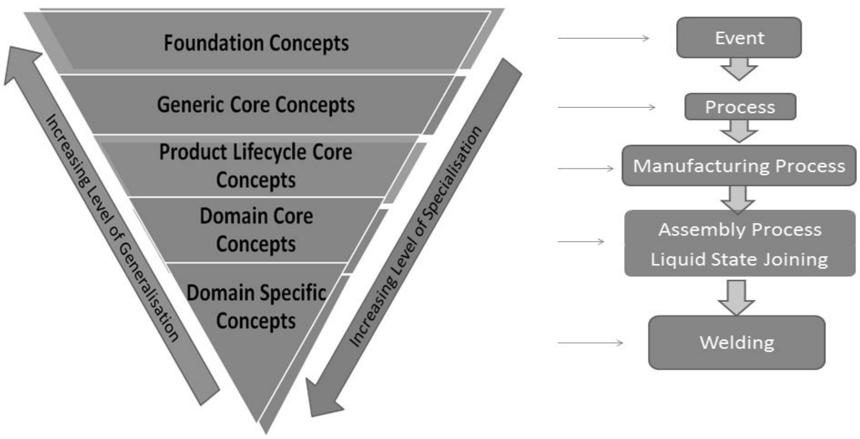


Figure 8Multiple levels within PLO with an example of Process concept

3.2.2 Knowledge Representation and Sharing Through Core Concepts

The requirement of a core or reference ontology is fulfilled through identification of core concepts. These core concepts have been developed by studying the existing ontologies, general literature on manufacturing processes together with different design and process planning software has been studied alongside with related developed ontologies. PLO has a set of shared or common concepts whose semantics are applicable for all the concerned domains addressed in this research. Hence, the knowledge that is signified by these concepts can easily be shared across the domains.

One of the most crucial requisite to establish a robust knowledge sharing system is the level of formalisation of the concept semantics and the most appropriate language to achieve this. From the review in the previous chapter it has been highlighted that textual definitions or text based semantics are open to misinterpretation, posing a barrier towards interoperability. Therefore, a more formal approach in the form of a heavyweight ontology which is capable of computationally capturing and interpreting the semantics is required. Rigorously defined rules and axioms are the most essential characteristics of heavyweight ontologies (Fürst, 2005). This is because they enable the system to interpret the semantics of the concepts and further apprehend to infer new knowledge. However, the choice of formalisation language dictates the level of expressivity and the inference capability. OWL was chosen as the formalisation language for PLO as it has the capability to process the content of the information rather than just presenting it. It's one of the widely used ontology formalisation language with high expressivity. The utilisation of SWRL further increases the inference capability of the language.

(Usman, 2012) had illustrated the representation of knowledge at meta-level and at the instance level (meta-meta level). However, requirements of this research were fulfilled through capture of meta-level knowledge from the core concepts and their relationships. This was achieved by declaring the concepts as classes and their relations as properties of the classes in the Protégé ontology editor. The classes act as variables that can have multiple instances and different types of relations defined between them. Rules and axioms defined on these classes and properties support the inference of new knowledge. The defined meta-level structure acts as the backbone for instance level knowledge structure. This is elaborated through the following example, a declaration *"Machining Facility has capability for Machining Operation"* is a meta-level knowledge structure where *Machining Facility* and *Machining Operation* are the concepts or classes, *has capability for* represent their relation or property. An individual level knowledge is represented through instances of these concept classes such as "*8Shop Machining Facility has capability for Scallop Machining"*, where *8Shop Machining facility* and *Scallop Machining* are the instances for the *Machining Facility* and *Machining Operation* respectively. Further instantiation of the instances are not possible.

It has already been stated that the primary route for knowledge sharing is the through the common set of concepts. Design, Machining, Welding and Inspection are different domains of the product lifecycle. There are several concepts which are common across all these domains and thereby provide the link to share this knowledge. However, there are several other concepts which are specific for the aforementioned domains. These are used to support the representation of domain specific knowledge and further act as route to extend PLO for a specific domain. Furthermore, these concepts form the basis to apprehend the consequences of changes between the domains. For example, *Product Feature* is a common concept across all the domains. *Design Feature* and *Machining Feature* on the other hand are not common across all. But as they are related with the common concept, they can be easily linked. The relation with other domain specific concepts and their formal definition allows the inference of new knowledge. This is illustrated through the following example in Figure 9; the *Machining Feature* concept has a relation with *Cutting Tool* and *Machining Process*, based on the *Design Feature*. This example shows that the statement if a design feature is defined to be a hole then the corresponding machining process required is a drilling process. Further, to perform the process a cutting tool in the form of a drill bit would be required.

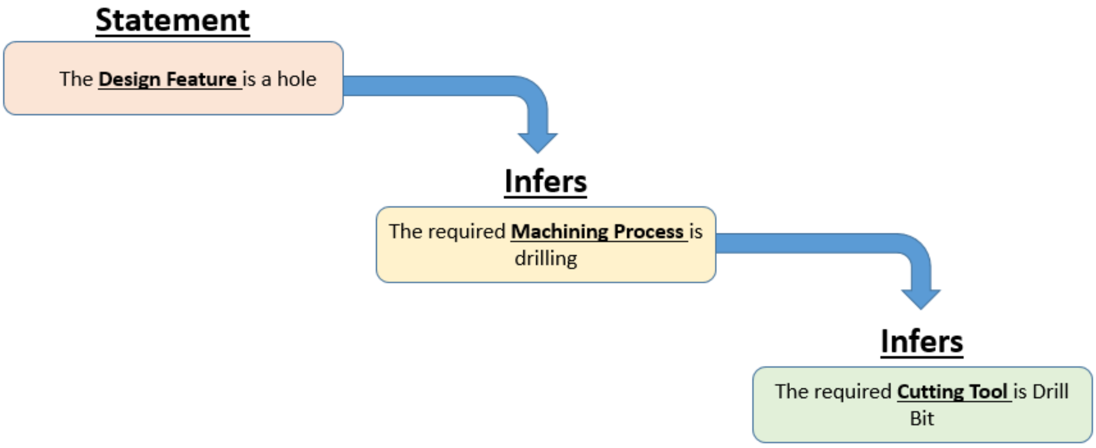


Figure 9 Example of rules to infer new knowledge

3.2.3 Modelling Process of PLO

The various ontology development methodologies have been elaborated in Section 2.5.3. The methodology proposed by (McGuinness, 2002) and (Blomqvist, 2008) are one of the most widely used methodologies for the domain of manufacturing (Chang, 2010) (Usman, 2013). Therefore, their methodology along with the additional step of formalisation of the concepts and semantic testing has been used for the development of PLO. The overall, implementation framework had been shown in Figure 2.

The problem apprehension for the development was carried out from literature survey and the industrial case study (step 1 and 2 on of the framework). This was followed by the UML modelling process in the following steps (step 3)

1. The identified core concepts are modelled as classes in a hierarchical form using a top down approach. E.g. the *Manufacturing Process* class which is a top level class was modelled first followed by its more specific *Assembly Process* class. Section 3.3.1 explains the practical elaboration of this step.
2. The attributes of the classes are defined through relationships, e.g. the attribute of the *Assembly Process* that it requires *Force* is defined through the relation *requiresForce*. Section 3.3.2 explains this step in practise.
3. The cardinality of every relationship is defined to capture the uniqueness of the relationships, e.g., 1 to 0.1 cardinality defined for *requiresForce* relationship, signifies that every *Assembly*

Process class can have a maximum of one unique relation with the *Force* class. Section 3.3.2 and Section 3.3.3 further elaborates on this step.

PLO is then formalised using the Web Ontology Language (OWL) by assigning rules and axioms as detailed in Section 3.3(step 3). A number of experimental verification methods have been used to confirm the ability to infer the consequences of design changes on the different manufacturing domains. It is one of the verification methods used for demonstrating the knowledge sharing capability of the model (step 4 and 5). The utilisation of PLO to consolidate the welding standards is another verification method for the correct semantic capture of the concepts (step 4 and 5) Furthermore, an industrial case study was modelled on PLO represent a real world scenario (step 6).

3.3 PLO Concepts

This section details the step 3 of the framework by exploring the various concepts of the PLO and detailing the investigation carried out to define these concepts. It follows the modelling process as explained in the previous section. The explicit definitions of the concepts along with the relationships between them have been further elaborated. Various concepts and terms related to product lifecycle have been studied from the literature, widely accepted standards and the product lifecycle management software tools. Some of the definitions and the meanings of the concepts have been adopted from published literature and standards while the others have been newly proposed. An industrial case study which included interviewing the domain experts was also instrumental in defining these concepts. PLO has been developed through specialisation and inter-association of core concepts. This association of the concepts are extended across the different specialisation levels. As previously described, PLO is a core or reference ontology and has its root at the foundation level. Therefore, it is imperative to define the most relevant foundation ontology and the corresponding concepts which would form the base for PLO. The foundational concepts of DOLCE ontology and Protégé have been utilised to fulfil this requirement because of their relevance as explained in Section 2.

In protégé, the basis or the foundation concept *Thing* is used to define everything. It is used as the upper most concept of the foundation ontology from which other concepts are specialised as shown in Figure 10. It is similar to other top level concepts of different foundation ontologies. Such as *Top* from the Upper

Level Ontology (ULO) used by (Usman, 2012) (Imran, 2013), *Entity* used by (Oberle, 2007), (Lambert, 2009). Specialising from here is where the foundational concepts from DOLCE ontology have been adopted. It must be highlighted here that the concepts which have been found to be relevant to define PLO concepts has only been adopted. It originates with *Particular*, the primary concept from the DOLCE upper ontology and describes the unique things of the universe in discourse (Imran, 2013). It is specialised into three sub classes, *Endurant*, *Perdurant* and *Abstract* as shown in section Figure 10. *Endurant* (also referred as continuants), are fundamentally physical objects which are wholly present at a given time and can further change with time. *Perdurant* on the other hand are occurrences which are only partially present in time during which they exist. *Abstract* represents the particulars which does not have any location. *Quantity* is a subsumed class of *Abstract* and further specialises into *Dimension* concept of PLO. *Endurant,* is specialised as *Physical Endurant* and further as *Physical Object* and *Feature*. They represent endurants with unity or spatial qualities. It essentially means that these concepts endure through time. However, *Feature* is a parasitic entity which is dependent on other entities. *Product* is an example of a PLO concept subsumed from *Physical Object.* *Event* is further classification of *Perdurant* that represents those concepts that unfold with time. *Process* is a concept of PLO, subsumed from *Event*. The foundational concepts are specialised in to the core concepts of PLO. Some of the key concepts are elaborated.

3.3.1 An Overview of Key Concepts of PLO

The primary purpose of the core concepts is to capture the knowledge from the domains of Machining, Welding, and Inspection and share it with Design. These concepts capture the various aspects from the aforementioned domains and represent them in a shareable format. Figure 10 in Appendix G, illustrates all the high level core concepts of the PLO. Some of the key concepts which are the prime drivers of knowledge sharing within PLO are *Product*¸ *Part*, *Feature*, *Resources, Setup, Platform, Facility, Operation, Process, Parameter, Dimension, Workpiece Orientation, Resource Tool Orientation, Operator Intervention, Method* and *Form.* It must be highlighted that these are the high level or parent classes of the concepts. A combination of these and some of their sub-classes further enable to capture and share knowledge, which are explained elaborately.

In this section the primary concepts which are required to enable the overall knowledge sharing are detailed. The specific concepts for capturing Assembly and Welding knowledge such as *Assembly Process, Joint Type, Mating Configuration* and their specialisation are explained in Section 4 while those for Manufacturing Operations such as *Manufacturing Method, Process Plan, Platform, Setup, Resource Tool, Resource Tool Orientation, Workpiece Orientation, Machine Tool, Fixture, Operator Intervention* and their specialisation are described in Section 5.

3.3.1.1 Product

The multidimensional interpretations of *Product* from different authors and commodities are owed to the differential perspective from single isolated domains. This varying implication can be understood from the different perspective of designers and manufacturing engineers. A designer is concerned with only the functional capabilities of the *Product* while the manufacturers are more interested in the required processes and tools to produce this. Few such examples of multifaceted definitions are, *"A thing or substance produced by a natural or artificial process"* (ISO-10303-1, 1994), *"An item designed, developed, manufactured and assembled to fill a functional need"* (Kim et al., 2006), "*A single item or an assembly of items required to be manufactured"* from Teamcentre (Siemens PLM Software) and it was referred as an assembled object by (Imran, 2013). Thus, a new definition has been proposed in this research which could be utilised generically across the domains. *Product* is defined as *"An unique item that fulfils one or more functional requirements for the end user which cannot be a part of a larger whole".* This definition can cater for single piece part manufacturing and assemblies. It further takes into account the perspective of supply chain. This concept has been introduced to capture its related knowledge and their corresponding implications in other domains. It is subsumed under *Physical Object*.

Product has further been specialised into *Discrete Product* and *Processed Product* as these encompass all of their forms. *Discrete Product* is defined as *"A product which is designed and developed to undergo the various stages of the Product Lifecycle and comprised of one or more parts"* such as automobile, furniture etc. *Processed Product* is *"A Product which is processed and cannot be broken down to its basic constituents"* such as milk, oil etc. *Processed Products* are beyond the scope of this research and hence *Discrete Product* has been further specialised as *Atomic Product and Compound Product. Atomic Products* are basically single piece part products and are defined as *"A Product which is made up of one* *discrete part.”*, such as a screw, bolt, spoon. *Compound Products* on the other hand are assembled products which are defined as *“A Product which is made up of multiple parts”*, such as engine, automobile.

3.3.1.3 Part

Part along with *Product* is perhaps one of those important concepts which have their implications across all the product lifecycle domains. (Usman, 2012) provided the most relevant classification of this concept, which has been adopted in this thesis. However, his definition was based on the ISO standard that stated *Part* as *“Discrete object that can come into existence as a consequence of a manufacturing process”*. It does not encompass the other product lifecycle domains and hence modified as *“Discrete object that can come into existence as a consequence of a manufacturing process which may be assembled with others to make a compound product.”* (Usman, 2012) further specialised *Part* into *Designed Part* and *Realised Part*, where *Realised Part* captures the different states of the part as shown by its specialisation in Figure 10. *Designed Part* represents a virtual part that is yet to be realised. The definition of *Realised Part* has been adopted from (ISO/TS-10303-1164, 2004), where it *“Represents a part that exists physically in the real world and whose properties can only be known by observation”*.

3.3.1.4 Manufacturing Facility

Manufacturing Facility and its related concepts have been found in various models. The origins of the concept can be traced back to the *Facility* model proposed by (Simpson, 1982). The extension of their model by (Zhao, 1999), (Molina, 1995) and (Usman, 2012) are perhaps the most comprehensive model that exists for product lifecycle. The additional facets from Zhao were that he considered *Manufacturing Facility* from high and low level perspectives. Usman further added its relation with the *Location*. Therefore from a high level perspective, *Manufacturing Facility* could be manufacturing shop, factory, cell or an enterprise while from an individual low level point of view it can be a single assembly station, machine etc. The definitions of the concept and the lightweight structure have been adopted from MCCO as shown in Figure 10. *Manufacturing Facility* has been explained to be *“a representation of an object that contains the Manufacturing Resources.”* It has been specialised *as Enterprise, Factory, Shop, Cell and Station*. Where, a *Station* represents *“a Manufacturing Facility consisting of a single working station”*. *A Cell* represents *“a Manufacturing Facility consisting of multiple Stations grouped to perform* *similar tasks".* A *Shop* represents *"a Manufacturing Facility consisting of multiple Cells grouped to manufacture Parts that are similar from a production perspective".* A *Factory* represents *"a Manufacturing Facility consisting of multiple Shops to produce a single Part, Product, set of Part, set of Products, or services"*. And finally, *Manufacturing Enterprise* represents *"a Manufacturing Facility consisting of multiple Factories grouped to contribute towards a common Product, set of Products, or services."*

3.3.1.5 Manufacturing Resources

Resource is a generic class of concept from which *Manufacturing Resource* has been specialised. The most appropriate definition of *Resource* was provided by (ISO 19115, 2003), stating it to be *"Assets or means that fulfils a requirement"*. The suitability of this definition is due to its ability to address a wide spectrum of domains beyond product lifecycle such as human, natural resources etc. However, the generic nature of this definition enforced the need for a more specialised *Manufacturing Resource* concept to capture the knowledge across the product lifecycle. It has been defined to represent *"the resources that enables a manufacturing process"*. Thus, it encompasses any form of resources that are required to produce a product or a particular feature of a product.

Several authors have provided different categorisation of Manufacturing Resources with different sets of subsumed concepts. The classifications proposed by (Leimagnan, 2006) in their MASON model, the machining ontology by (Semere, 2007) and within the MCCO by (Usman, 2012) were found to be most relevant for this research. These models have been modified to provide a more generic set of concepts that can capture the knowledge from different product lifecycle domains. Therefore, *Manufacturing Resource* have been further specialised as *Machine Tool, Resource Tool* and *Fixture*. These concepts are perceived to capture all the types of resources which are required to carry out any manufacturing process. The definition of *Machine Tool* by (Semere, 2007), (Usman, 2012) and (ISO-16100-1., 2009) were very specific for the machining domain. Hence, the newly proposed definition states Machine Tool to be a "*representation of Manufacturing Resource on which the Resource Tool, Workpiece and Fixture(s) are arranged."* A machining centre, coordinate measuring machine, welding bode can be different types of *Machine Tool*. *Resource Tool* on the other hand is described to be those set of resources that are used to carry out the corresponding manufacturing process. For example, a cutting tool is used for machining process, while a welding torch used for a welding process. Finally, *Fixture* denotes *"the Manufacturing Resource which is used to hold the Workpiece or the Resource Tool or the both."* .It must be noted here that Human Resources has not been designated as a specialised concept of Manufacturing Resource as proposed by aforementioned authors. It has been subsumed under *Resource* as their semantics prevails beyond the realms of product lifecycle, e.g. human resource employees of university.

3.3.1.6 Manufacturing Process

The importance of *Manufacturing Process* concept is highlighted from the requirement of capturing knowledge from multiple domains. Each of these domains comprise of their own variety of *Manufacturing Process* that requires to be addressed. *Manufacturing Process* is specialised from the more generic *Process* where it is defined as *"An event or series of events resulting in a change of state."* Here state is *"a condition or way of being that exists at a particular time."(Cambridge)*. Therefore, *Manufacturing Process* is newly defined as *"Structured set of activities or operations that is performed upon an object and contributes towards converting it from a raw material or a semi-finished state to a state of further completion."*

The further specialisation from *Manufacturing Process* paves the way into the Domain Concept level as shown in Figure 10. Here there are three specialised concepts based on the perspective of this research work. They are *Machining Process, Assembly Process and Inspection Process*. The further exploration and specialisation of *Assembly Process* are detailed in Section 4. Here, *Machining Process* has been defined as *"Processes that consist of the removal of material and modification of the surfaces of a workpiece after it has been produced by various methods". Assembly Process* has been defined as *"Process by which a group of components are brought together under specific mating conditions to form a unit."* And, *Inspection Process* has been defined as *"Process for careful examination or scrutiny to determine conformity."*

3.3.1.7 Manufacturing Operation

Manufacturing Operation is one of the key concepts within PLO which is required to capture and share the knowledge pertaining to manufacturing sequences. It is a specialisation of the more generic concept *Operation*. This concept along with their newly proposed specialised concepts including their novelty is explored in Section 5.

3.3.1.8 Feature

Feature is the most critical concept of this research work, as it provides the route to share knowledge across different product lifecycle domains. More specifically, it is the key concept through which the manufacturing knowledge is shared to design. According to Oxford dictionary, *Feature* has been described as *"A distinctive attribute or aspect of something"*. (Kim et al., 2006) explained it to be a region of interest within a part or an assembly and are defined by attaching some "attributes". Imran adopted the definition from (Pratt, 1985), stating *Feature* to be a *"region of interest on the surface of a part"*. Several other similar definitions have been proposed by other authors. It could be noticed from all these definitions that *Feature* will always have some attribute of interest. This entails that *Feature* is quite generic in nature with universal semantics. Thus, the definition proposed by (Usman, 2012) of *Feature* being *"anything having a particular attribute of interest"* was found to be most appropriate for this research. However, its semantics can be varying from a generic perspective to product lifecycle specific. For example, a generic notion of *Feature* could be a smile, intelligence, colour of the eyes etc. While, more specifically for product lifecycle domains it could be a stress relieving feature, turning feature etc. Hence, *Feature* is further specialised into several concepts or subclasses. The hierarchical structure of the *Feature* concepts is shown in Figure 10.

The specialised classes of *Feature* were required to ensure the proper capture of varying depth of semantics from the different specified domains. The *Form Feature* specialisation followed by the subsumption of the *Product Feature* has been adopted from the model proposed by (Usman, 2012) and (Imran, 2013). However, the further classifications have been newly proposed. The notion of *Form* has been agreed because any type of *Feature* will always have a *Form* and further related to a *Product*. These concepts have been described by various authors to model their own domain specific manufacturing ontologies. For example, (Rosen, 1993) describes *Form Feature* from a machining perspective as *"features producing volume",* while (Roller, 1989) had a bit more generic view and stated it to be *"features that relate to shape or form of the part".* However, PLO and its core concepts are more generic than specific domains. Thus, the definition proposed by Usman where *Form Feature* is described as *"a Feature which has Form as its required attribute of interest"* was found to be appropriate as the generic semantics encompasses multiple domain perspectives. A door, window, eyes, wing mirrors are some example of *Form Features*. Similarly, *Product Feature* has been proposed to be defined as *"Feature that is associated to a Product"*. This is because every feature of a particular product is unique for itself such as cooling holes on turbine blades, handle of a mug. These are shown in Figure 11.

Figure 10Examples of Feature and its uniqueness for different products

The further specialisation from *Product Feature* leads to *Design Feature, Manufacturing Feature and Resource Feature* concepts. Here, *Resource Feature* denotes the features of resource tools that carry out any manufacturing process to create a product feature. It must be noted that *Resource Feature* has been subsumed under the *Product Feature* because the utilisation of a resource feature is always associated with a particular product. An example of a resource feature could be the "Tip" of a cutting tool or the "socket" of a nut runner as shown in Figure 25.

Figure 11Examples of Resource and Design Features

Design Feature is design domain specific concept that is dependent on the functional aspect of the *Product Feature*. It is defined as *"a Product Feature that has Design Function as a defining attribute of interest"*. Slot feature is an example of *Design Feature* as shown in Figure 12. Similarly, *Manufacturing Feature* has been defined as "*Product Feature that has a Manufacturing Process as an attribute of interest"*. The specifics of the multiple domains within manufacturing are captured through its further specialisation into *Machining Feature, Assembly Feature and Inspection Feature*. Each of these concepts is defined based on the corresponding process as an attribute of interest. Such as, *Machining Feature* is defined to be a *"Feature that has Machining Process as an attribute of interest".* And the others are defined respectively.

The *Assembly Feature and Inspection Feature* have been newly specialised to capture the very specifics of the corresponding domains. *Mechanical Assembly Feature* concept is used to capture the feature knowledge which is entirely pertaining to mechanical assembly processes, such as riveting, bolting etc. However, in order to capture this knowledge with higher granularity, two more specialised concepts *Male Assembly Feature* and *Female Assembly Feature* have been portrayed. This specialisation is based on the understanding that every mechanical form of assembly or joint requires a *Male Assembly Feature* and *Female Assembly Feature* to mate with each other. For example, in a cylindrical joint, the shaft is the *Male Assembly Feature* while the hole is a *Female Assembly Feature* which mate together to form the joint. The *Non Mechanical Assembly Feature* concept captures the knowledge of all the features associated with non-mechanical assembly processes such as welding, brazing etc. It is further classified as *Continuous Non Mechanical Assembly Feature* to capture the feature related information for processes that does not involve any detachment of the *Resource Feature* from the *Product Feature* during the entire cycle. Seam, groove and fillet weld features are some examples. Similarly, the other classification is *Intermittent Non Mechanical Assembly Feature*. It captures the feature related information for processes that involves detachment of the *Resource Feature* from the *Product Feature* during the entire cycle. Spot weld feature is an example of such feature. The *Inspection Feature* specialisation is to capture its two distinct types, *Geometrical Inspection Feature and Attribute Inspection Feature. Geometrical Inspection Feature* captures all the geometrical inspection characteristics which could be *Fundamental* (straightness, roundness and flatness) that are based on the fundamental measuring parameters. It can also be *Compound* (cylindricity, conicality, concentricity) which is based on the combination of different measuring parameters.

3.3.2 Relationships between Concepts

The previous section provided an overview of the primary classes of concepts within PLO. More, specifically the high level concepts which are crucial to share the knowledge from multiple product lifecycle domains to design had been described. Figure 10 shows only the hierarchical relationship of concepts i.e. their specialisations. However, the model requires additional relationships defined between the concepts to ensure the complete sharing of knowledge. These additional relationships are described in this section.

Figure 13 shows the UML model with their intra and inter category relationships between the concepts. It must be observed that all the relationships have been defined at the generic level for the super classes. This is based on the understanding that the subsumed classes inherit all the attributes of their super classes, including all of their relationships.

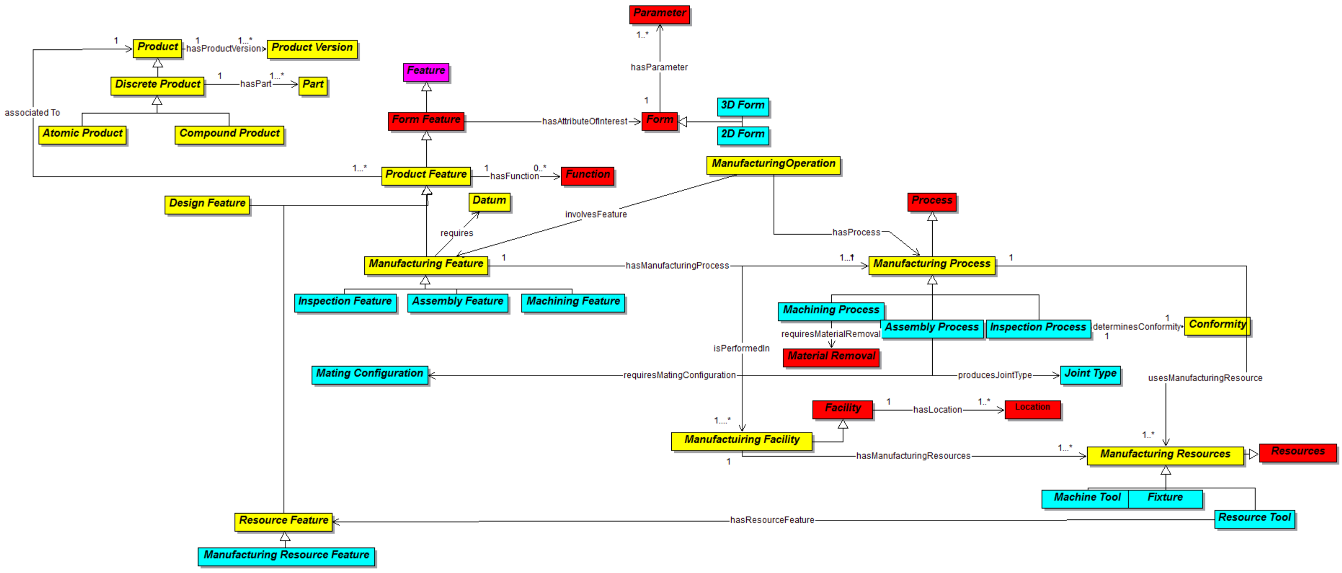


Figure 12 UML Model of PLO to share knowledge

The knowledge associated with the concepts is captured with the help of these relationships. These are further used to define rules, axioms for formalisation (discussed in Section 3.3.3) and for querying the knowledge base. From the above Figure 13 it can be observed that these relationships basically define the associations between the key terms from the concept definitions (as explained in Section 3.3.1). For example, *Discrete Product* has been defined as to be comprised of one or more *Parts.* Hence, the relation *hasPart* has been defined between *Discrete Product* and *Part* classes. The notion of every *DiscreteProduct* has at least one *Part* is described through the cardinality of 1 to 1* on the relationship. Similarly, the relation *associatedTo* with cardinality 1…* to 1 explains that one or many *Product Feature* is associated to only one *Product* as shown in Figure 26. To capture the knowledge about different versions of the same product, relation *hasProductVersion* has been defined between *Product* and *Product Version.*

A crucial aspect of *Form Feature* is its requirement to have *Form* as it's attribute of interest. Therefore, the *hasAttributeOfInterest* relation has been defined between the two classes. The *hasParameter* relation with cardinality 1 to 1… * between *Form* and *Parameter* illustrates that every form has at least one parameter defined. It should be noted that although relation *hasFunction* has been defined at the generic *Product Feature* class with *Function* but this relation only holds for *Design Feature* class. This scenario is addressed through the cardinality 1 to 0..*, which would mean that some of the product features may not have any function as in case of *Manufacturing Feature.* The class specific constraint is achieved through formalisation rules and axioms as described in Section 3.3.3. The form of any feature requires some parameters to describe them. These are primarily the dimensions and other attributes and are captured through the *hasParameter* relation between *Form* and *Parameter* classes.

The definition of *Manufacturing Feature* stated the requirement for an association with *Manufacturing Process.* This is illustrated through the *hasManufactruingProcess* relationship between the two classes with cardinality of 1 to 1..*. The subsumed class specific constraints are defined through the rules and axioms. An additional relationship with *Datum* has been defined through the *requiresDatum* relation as every manufacturing feature needs a defined datum to carry out the operation. Therefore, a similar type of relation was defined between *Manufacturing Operation* and *Manufacturing Feature.* The *involvesFeature* relation between the aforementioned classes signifies that every manufacturing operation involves at least one type of manufacturing feature. Further, the corresponding relation with *Datum* reveals that every feature which is involved in the manufacturing operation will have datum defined. It has been stated in the previous section, that every *Resource Tool* has a particular unique *Resource Feature*. Hence, it is captured through the *hasResourceFeature* relation between the aforesaid classes.

Manufacturing Operations is associated with the *Manufacturing Process* class through the *hasManufacturingProcess* relation. This denotes that every manufacturing operation carried out involves a certain type of manufacturing process. Furthermore, the *Manufacturing Process* has two crucial associations with *Manufacturing Facility* through *isPerformedIn* relation and *Manufacturing Resources* via the *usesManufacturingResources* relation. These relations states that every manufacturing process uses one or more manufacturing resource and is carried out in a particular manufacturing facility. The concerned *Manufacturing Facility* also houses the *Manufacturing Resources*, as defined through the *hasManufacturingResources* relationship. The geographical location of the *Manufacturing Facility* is captured through the *hasLocation* relation between the super class *Facility* and *Location.*

The specialised concept classes of *Machining Process*, *Assembly Process and Inspection Process* have their own specific relations to define themselves. The necessity of an *Assembly Process* to produce a *Joint Type* and requiring a *Mating Configuration* is achieved through the *producesJointType* and *requriesMatingConfiguration* relations respectively. Similarly, the *determinesConformity* relation is assigned to *Inspection Process* with *Conformity*, as the prime objective of inspection processes is to determine the conformance of the feature. Also, the *Machining Process* class has a relation *requiresMaterialRemoval* with *Material Removal* class, as every machining process involves some form of material removal.

These relations illustrated in Figure 13 shows an abstract view of the complex relations within the proposed PLO. More specifically these are high level concept classes and their relationships. They are required for multi domain knowledge sharing. The specific relations for *Assembly Process* and *Manufacturing Operations* knowledge capture have been further explored in details in Sections 4 and 5 respectively.

3.3.3 Formalisation of PLO

The final step of the modelling process is the formalisation of the identified concepts and their relations. This section describes the formalisation of the primary core concepts which have been described above. Based on the discussion in Section 2.5, the Web Ontology Language Description Logic (OWL DL) has been used as the formalisation language. OWL DL is an extension of RDF and RDFS, providing semantics with regards to explicitly representing complex constraints. It uses the syntax of XML and RDF. They provide the essential syntax and semantics which are required for knowledge modelling. These are primarily in the form of concepts, relations between the concepts and the logical constraints which they satisfy. The modelling of PLO follows the following fundamental elements of modelling in OWL, i.e.

1. Namespaces
2. Classes (Concepts) and Relations (Properties)
3. Rules, Restrictions and Axioms

3.3.3.1 Declaration of Namespace

In OWL DL, namespaces act as identifiers to represent the ontological entities and further address the different contexts. They provide overall indications regarding the background of the vocabularies used. These provide a means to unambiguously interpret identifiers and make the rest of the ontology presentation much more readable (W3C, 2004). The following declaration shows the method of declaring the ‘PLO’ identifier for the proposed ontology.

xmlns="http://www.owl-ontologies/PLO.org#"

xml:base="http://www.owl-ontologies/PLO.org"

3.3.3.2 Classes and Property Declarations

The concepts are declared in OWL DL as classes while relations are declared as properties. Therefore, the most fundamental concepts are declared as the root classes of the taxonomic components. The directive “<owl:Class rdf:ID >” is used to define the class, while the directive “<rdfs:subClassOf>” is used to capture the subsumption relations. Here the subclasses and their individuals inherit the properties and other restrictions of their super classes through the subsumption relation. The declaration of *Product* and their related sub classes is shown below. This belongs to the ‘PLO’ namespace. The ‘PLO’ ontology imports the more generic ‘Foundation’ ontology as *Product* is a specialisation of the generic class *Physical Endurant*. The prefix of the concepts denotes the specialisation level they exist and further their dominion. For example, the prefix *ProductLifecycleCore* denotes that the concerned concept belongs to the *Product Lifecycle Core* level of PLO. There are 2 types of binary properties that can be declared in OWL DL. These are Object Properties which are used to define the relations between instances of two classes. They are declared using the “owl:ObjectProperty” directive. The other form of properties is Data Properties which are used for relations between instances of classes and RDF literals, XML Schema datatypes i.e. for any numeric functions. It uses the directive “owl:DatatypeProperty” to define the relation. The directionality of the relationships are specified by their domain (rdfs:domain) and range (rdfs:range) respectively. The snippet of the code for implementation is shown in Appendix C.

3.3.3.3 Declaration of Restrictions, Rules and Axioms

The restrictions in the OWL DL are primarily for defining the constraints and axioms which infuses semantic enrichment to the ontology. These axioms provide the consistency checking of the ontology which includes the assertion of individuals in to the Knowledge Bases (KB) created in OWL. Together with different rules, they act as inference mechanisms to deduce new knowledge based on the restrictions and identify the equivalency relations among the classes.

OWL DL on its own provides two types of restrictions which are 'necessary conditions' and 'necessary and sufficient conditions'. The 'necessary conditions' are used to support the creation of a primitive class by specifying an anonymous super class of a named class. A typical method of inducing such restriction is portrayed in the following code. A necessary condition is placed on *Liquid State Joining* where it ensures that the semantic consistency of the instances has relation with *Melting Temperature* via *reachesMeltingTemperature* relation. It also ensures that any defined class with the above relation would be sub classified under *Liquid State Joining.*

The 'necessary and sufficient conditions' are used to support the creation of a defined class by specifying an anonymous super class of a named class. Further it infers that any other class which has similar restrictions as its equivalent class. Few example codes are shown in Appendix C. These types of restrictions have been further explored in Section 4 where they have been extensively utilised.

In order to enhance the semantic rigour and further implement complex rules, OWL offers the Semantic Web Rule Language (SWRL). It is an extension of OWL that permits complex rule definitions and advance reasoning over the concepts. These rules can further be used to infer new knowledge. The syntax followed in defining the rules are in the form of antecedent-consequent pairs. This is a crucial requirement to model the complex scenarios for knowledge sharing. The syntax for defining rule is shown in Appendix C.

An important notable aspect of OWL which is required to be taken into account during the assertion is that OWL works on Open World Assumption (OWA). This implies that the system will not be able to infer something unless it is explicitly specified. E.g. If something is specified to be not true does not mean that it would be false rather it is simply inferred as unknown. Therefore, the individuals that are different have to be explicitly defined. This has been practically demonstrated through the experimental verification in Section 4.

3.4 Route to Knowledge Sharing From Machining, Assembly, Welding and Inspection to Design

One of the objectives of PLO is to act as a semantic base to develop application specific ontologies. Additionally and more crucially it is required to provide the route for knowledge sharing. In the context of this research, PLO supports the development of application specific design and manufacturing ontologies. Figure 11 in Section 3.2 showed an abstraction of developing application specific design and manufacturing ontologies from PLO via the domain ontologies. This section further elaborates on the ability of building application specific ontologies from PLO and further explains the route to share knowledge.

Based on the framework modelled in Figure 11, the *Feature* concepts have been exploited to develop the application specific design and manufacturing ontologies as shown in Figure 14. The scenario is elaborated based on the case study. An “Aero Engine OGV Design Ontology” and an “Aero Engine OGV Manufacturing Ontology” has been used as application specific ontologies. These ontologies capture the design and manufacturing views of the OGV assembly which has been described previously in Section 3.1. The specific views are captured as instances of the application specific ontologies as shown below.

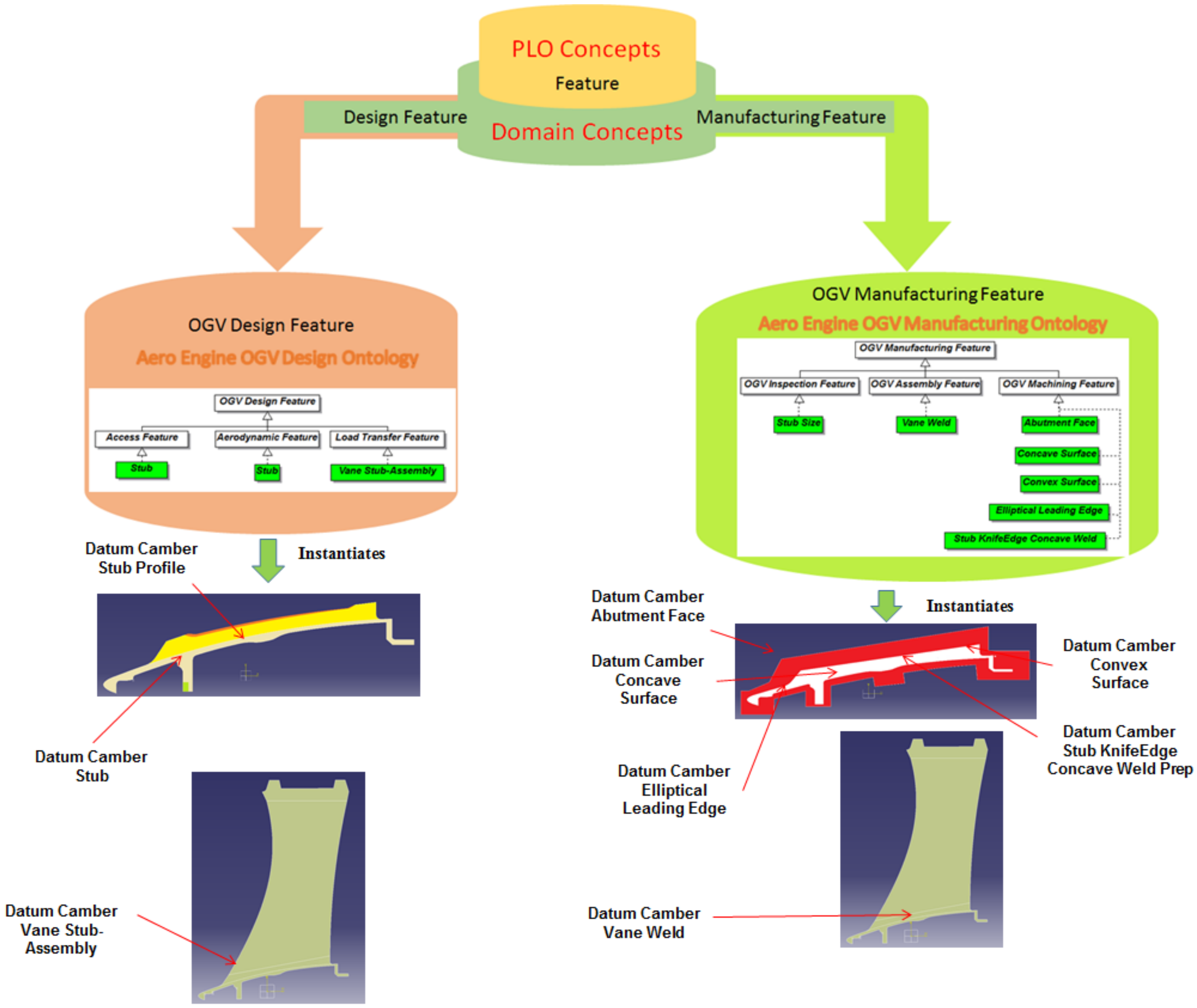


Figure 13A Lightweight Representation of Development of Application Specific Ontologies from PLO

From Figure 14, it can be observed that the application specific ontologies are supported by two main domain concepts (*Design Feature* and *Manufacturing Feature)* which in turn are supported by the concepts from PLO (*Feature)*. The detailed lightweight model showing the complete structure from PLO to the application specific ontologies (including the instances) is shown in the UML Figure 15. This illustrates the development of the application specific ontologies from PLO through the domain layer. The bottom part of Figure 14 and 15 shows the instantiation of the application specific concepts to capture the design and manufacturing views. The core concepts from the PLO have been used to develop the application specific ontologies. This implies that the application specific ontologies acknowledge PLO. Moreover, the commitment to PLO ensures semantic consistency along with the route to share knowledge. Although, the example shows the utilisation of the *Feature* but other concepts from PLO such as *Form, Product, Manufacturing Process etc.* and relations such as *hasFunction, hasAttributeOfInterest* etc. have been utilised as well. This ensures that the semantic integrity is maintained throughout, from PLO till the application specific ontologies.

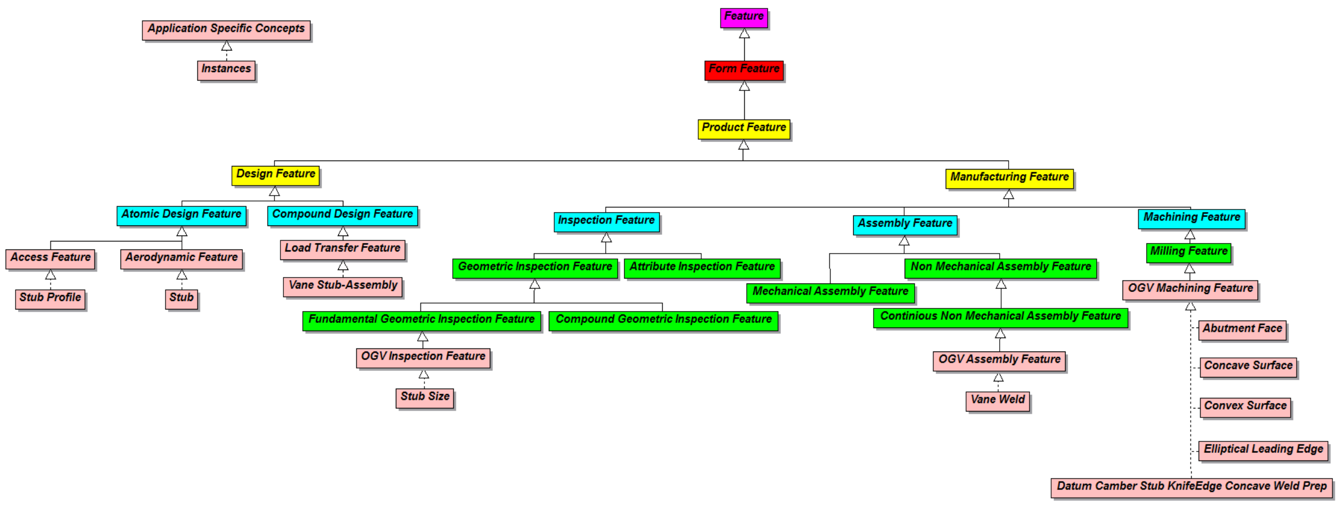
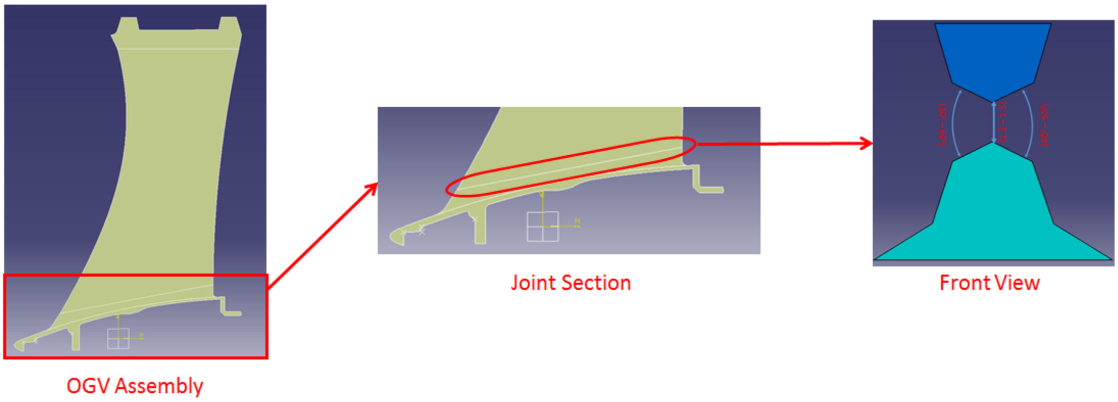
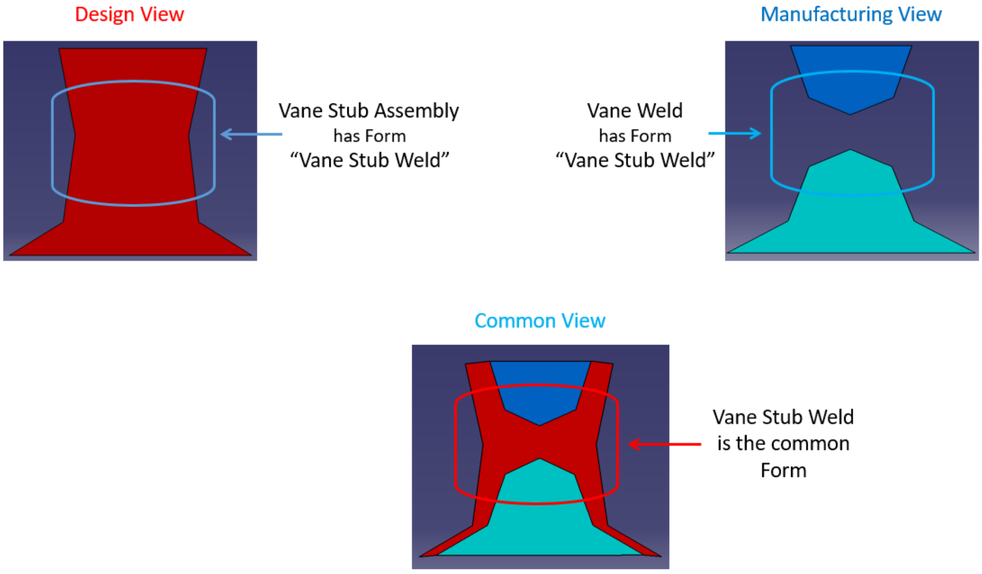


Figure 14UML Diagram for developing Application specific concepts from PLO

The development of application specific design and manufacturing ontologies from PLO is the first step that allows the knowledge to be shared with each other. It has been shown above that the design and manufacturing features can be linked together through *Feature* concept. For example, the different OGV Design features can be related to the OGV Manufacturing features using the *Product Feature, Form Feature and Feature* as shown in Figure 16. The two crucial concepts that establish the key link between the OGV design and manufacturing features are: *Product Feature* and *Form*. Therefore, if different manufacturing and design features are related to the same product then the knowledge can be shared by linking them through the *Product Feature* concept. However, the information pertaining to a certain feature can be obtained from their corresponding *Form*. This is explained through the following example.



(a)



(b)

Figure 15Relating Design and Manufacturing Features via Common Forms

Figure 16 shows a case where the consequences of changing a design on manufacturing are required to be identified. It shows the OGV Assembly, but focused on the joint section between the Vane and the Inner Ring. Figure 16 shows the front view of the joint. In this scenario, the concerned *Design Feature* is the “Vane Stub-Assembly”. The consequences in manufacturing from changing this feature can be found only when the corresponding *Manufacturing Feature* for the “Vane Stub-Assembly” is identified. This is carried out by first identifying the *Form* of the *Design Feature.* The next step is to identify the *Manufacturing Feature* which has common or partially overlapping form. In this case the *Form* of the *Design Feature* was identified to be “Vane Stub Weld”. And consequently, based on the relation defined between “Vane Stub-Assembly and “Vane Stub Weld” through their common *Form*, the relevant

Manufacturing Feature was inferred to be "Vane Weld". Thus, the associated knowledge of the form "Vane Stub Weld" permits determining its design-ability and manufacturability. The manufacturing knowledge associated with *Manufacturing Feature* is found in their manufacturing method. Their relation was shown in the Figure 14 UML diagram. Therefore, based on the identified relation between "Vane Stub-Assembly" (*Design Feature)* and "Vane Weld" (*Manufacturing Feature)* through the common form "Vane Stub Weld", the manufacturing knowledge can be shared to design.

The knowledge associated with "Vane Stub Weld" which is required to be fed back to design i.e the critical dimensions are shown in the front view of Figure 16(a).

1. The root gap between the vane and the stub must not be less than 1.2mm or more than 1.5mm in order to carry out the welding process
2. The concave side groove angle should be between 20^o and 30^o to carry out the welding process.
3. The convex side groove angle should be between 30^o and 40^o to carry out the welding process.

The capture of the above knowledge and sharing it back to design has been experimentally verified in the next section.

3.5 Experimental Validation

This section illustrates the different experimental investigation carried out to validate PLO. These experimental tests are used as a verification method for the previously described attributes of PLO. The verification methods validate the research hypothesis and highlight the novel aspects of the proposed model, PLO.

3.5.1 Methodology for Experimental Validation

In order to conduct the experiments that validates this research, the following methodology in Figure 17 have been followed.

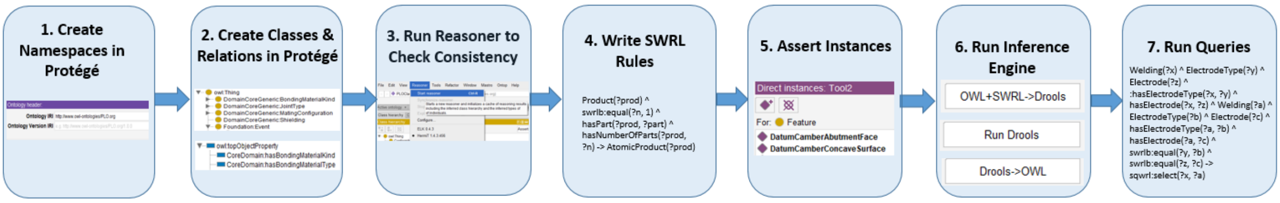


Figure 16Overview of the Methodology for Experimental Validation

As described previously, OWL has been used as the ontology development language for PLO. And the Protégé ontology editor is used as the development tool. The definitions of the classes and the OWL syntax of their formalisation have been described in the previous chapters. However, the Protégé editor allows a user friendly environment to define the classes and their relationships. The above Figure 17 explains the steps to carry out this procedure.

The first step for implementing the experimental tests is to create the namespaces in Protégé. The namespaces helps in defining the different specialisation levels. Step 2 involves the creation of different classes as a hierarchical structure within each namespace. Further, the different relations between these classes are also defined in this step. The consistency of the ontology is checked in the next step using the in-built reasoners of Protégé. Once the ontology has been deemed consistent, different rules and axioms are defined using the SWRL code in Step 4. This is followed by assertion of instances into the ontology. The assertion allows identifying any missing semantics which are required to be defined. Step 6 is where the inference engine is activated that applies all the defined rules and axioms onto the ontology. The final step involves using SQWRL to query the ontology for its evaluation. This methodology has been used for all the experimental validations and the case study.

3.5.2 Verifying the Semantic Integrity of PLO and Capturing the Knowledge from Multiple Domains

The primary objectives of this experiment are as follows

1. To verify the semantic integrity of PLO i.e. the model is devoid of any semantic inconsistency.
2. To verify the semantic capture of PLO
3. To verify that PLO is able to provide a route to share knowledge

It must be highlighted that this experimentation primarily focuses on the high level knowledge sharing aspect of PLO from multiple domains. The first step is to build PLO in the Protégé ontology development editor before the experimentation can be carried out. The formalisation methodology explained in Chapter 3 has been used to create the classes and relations in Protégé. The complete taxonomy of all the concepts starting from the foundation level to the application specific domain is illustrated in Figure 18. The classes belonging to each specialisation level has been imported from its predecessor. The context of the concepts depicts the different levels of the ontology they belong to. The change of the context reveals how the different concepts have been inherited from each other and linked with the concepts from PLO. Figure 18 further shows the development of the application specific ontologies from PLO. The successful capture of concept semantics is invariably a verification of the need for specialisation levels to capture the varying depth of meanings. This same methodology has been used for the Experiment 2, 3 and the case study.

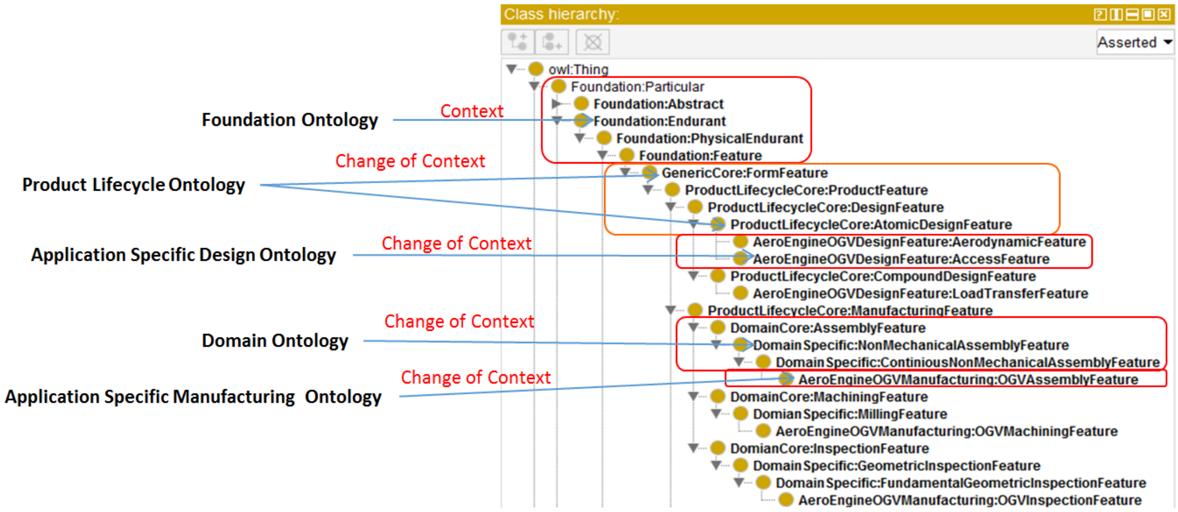


Figure 17Defined Classes of PLO, the specialisation levels and the application specific ontologies in Protégé

With the ontology being build and loaded in the Protégé environment, different experiments were carried out to verify the above mentioned objectives. Before the experiments were conducted, the following instances have been asserted into the knowledge base for their corresponding classes as shown in Table 2. It must be noted that these instances are asserted to their super classes. The reasoner inference engine automatically classifies them to their individual specific classes using the rules and axioms. However, these instances can also be directly asserted to their corresponding classes.

Table 2Asserted Instances and their Classes

Classes	Instances
Product	OGV Assembly
Manufacturing Process	Inner Ring Machining OGV Weld Stub Measure
Product Feature	Stub Stub Profile Vane Sub-Assembly Vane Weld Abutment Face Stub Size
Form	Stub Form Stub Weld Prep Form Stub Joint Vane Stub Weld Stub Profile Surface

1. With the knowledge base being populated with the instances, the semantic integrity of PLO is verified by reasoning the ontology. The Pellet Reasoner has been used for verification. The reasoner returned no error which is shown in Figure 19. Therefore, the semantic integrity of the model is preserved.

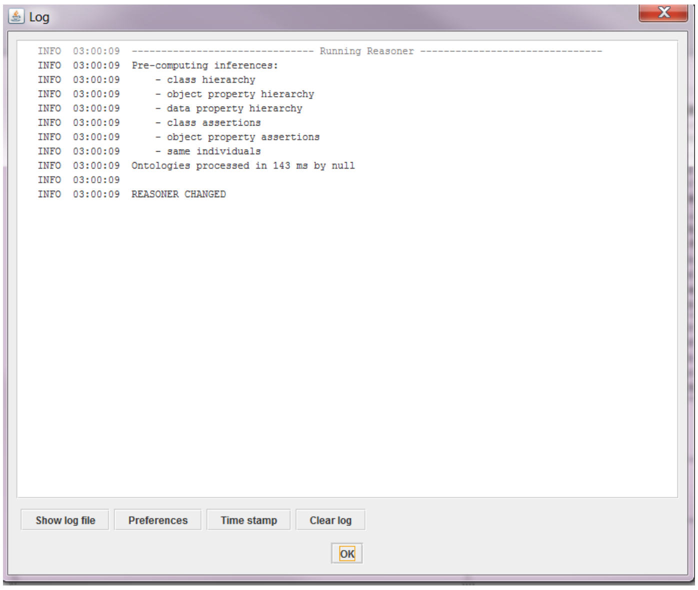


Figure 18Results from invoking the Reasoner

2. The successful capture of semantics by PLO is verified through three different aspects, such as

    a. Firstly, based on the rules and axioms the reasoner classifies the asserted instances to their corresponding classes. This is shown through the classification of the Design and Manufacturing Feature in the below Figure 20.

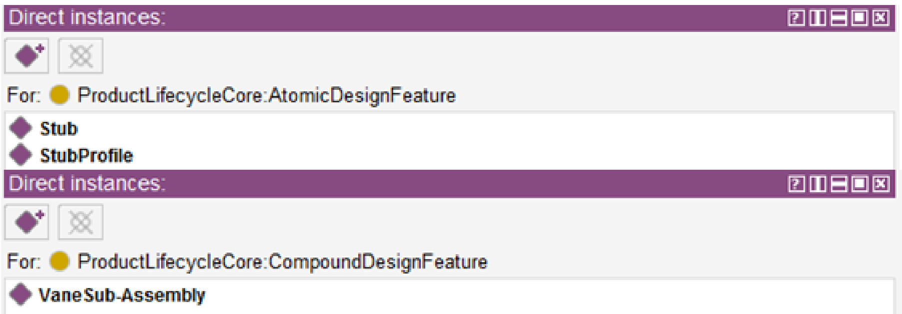


Figure 19Classification of the different Features

    b. Secondly, the successful capture is portrayed by highlighting the missing and captured semantics of the instance 'Stub' from above. The missing semantics are marked with red in Figure 21, which prompts the user to populate these values.

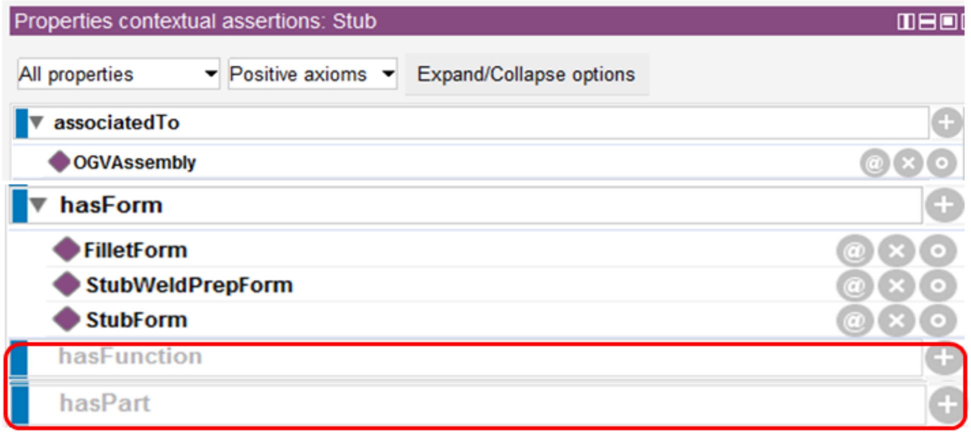


Figure 20 Assertion of Instances with Missing Semantics

c. Furthermore, the assertion of instances with inconsistent semantics results in error as shown below. In the following scenario, the instance 'Stub' was asserted as a *Design Feature* with a *Manufacturing Process*. It results in an error as shown in Figure 22, along with its explanation. This is due to the inherent semantics of the class defined through the assigned properties. The semantics of the *Design Feature* class requires its instances not to have any *Manufacturing Process* defined. Therefore, the capture of semantics of the proposed model and the corresponding definitions of the concepts is verified.

An Error Occurred During Reasoning

Inconsistent Ontology Exception: Cannot do reasoning with inconsistent ontologies!

Reason For Inconsistency: *Individual http://www.owl-ontologie/PLO.org#Stub is forced to belong to class all (http://www.owl-ontologies/PLO.org#hasManufacturinfProcess, not (http://www.owl ontologies/PLO.org#ProductLifecycleCore: ManufacturingProcess))and its complement "*

Explanation

Explanation for :owl:Thing SubClassOf owl:Nothing

Stub hasManufacturingProcess InnerRingMachining
hasManufacturinProcess Range ProductLifecycleCore:ManufacturingProcess
Stub Type ProductLifecycleCore:DesignFeature
ProductLifecycleCore:DesignFeature EquivalentTo ProductLifecycleCore:ProductFeature
and(not(hasManufacturingProcess some ProductLifecycleCore:ManufacturingProcess))
and (associatedTo some ProductLifecycleCore:Product)
and (hasForm some GenericCore:Form)
and (hasFunction some GenericCore:Function)
and (hasPart min 1 ProductLifecycleCore:Part)

Figure 21 Error due to Assertion of Instances with inconsistent semantics

3. As discussed in Section 3.4, the capability of PLO to provide a route to share knowledge can be verified through identification of the *Feature* that relates the different domains of manufacturing with design. Furthermore, the link between the *Design Feature* and the *Manufacturing Feature* is related via their Common or Overlapping Forms using the *Form Feature* concept as described in Section 3.4. Therefore, the knowledge base was queried to identity the common or the overlapping forms using the following SQWRL query. The result of the query is shown in Figure 23 below.

<u>SQWRL Query:</u>

DesignFeature(?DesignFeatures) ^ ManufacturingFeature(?ManufacturingFeatures) ^ Form(?CommonOrOverlappingForm) ^ hasForm(?DesignFeatures, ?CommonOrOverlappingForm) ^ hasForm(?ManufacturingFeatures, ?CommonOrOverlappingForm) -> sqwrl:select(?DesignFeatures, ?CommonOrOverlappingForm, ?ManufacturingFeatures)

DesignFeatures	CommonOrOverlappingForm	ManufacturingFeatures
StubProfile	StubProfileSurface	StubSize
Stub	StubForm	ConcaveSurface
Stub	FilletForm	EllipticalLeadingEdge
Stub	StubWeldPrepForm	StubKnifeEdgeWeldPrepHeight
Stub	StubForm	ConvexSurface
Stub	StubWeldPrepForm	StubKnifeEdgeConcaveWeldPrep
Stub	StubWeldPrepForm	StubKnifeEdgeConvexWeldPrep
Stub	FilletForm	EllipticalTrailingEdge
Stub	StubWeldPrepForm	AbutmentFace
VaneSub-Assembly	VaneStubWeld	VaneWeld

Figure 22 Identification of route to share knowledge

From the above Figure 23, it can be seen that the three different design features *Stub, Stub Profile and Vane Sub-Assembly* are related to their corresponding manufacturing features via the different common or overlapping forms. It can be noticed that *Stub* design is related to multiple manufacturing features. This signifies that the *Forms* are basically common or overlapping and not the exact same. The identification of the common or overlapping forms establishes the route to share the manufacturing knowledge to design. Therefore, the manufacturing knowledge for the corresponding *Form Feature* needs to be retraced from the knowledge base. Figure 24, shows the critical parameters that dictate the manufacturability of the Design Features. This information is established through the common or overlapping forms as shown in the above Figure 23. Furthermore, the restrictions on the parameters signify the limits the designers need to adhere from the manufacturing perspective. Such a scenario is portrayed through a case study as described in Section 3.5.3. Thus, this shows the capability of PLO to establish the route to share the knowledge from the domains of machining, assembly with welding and inspection to design.

DesignFeatures	CriticalParameters
:Stub	KnifeEdgeConvexWeldPrepAngle
:Stub	AbutmentFaceLength
:Stub	VaneStubDatumCamber
:Stub	KnifeEdgeConcaveWeldPrepAngle
:Stub	StubKnifeEdgeHeight
:Stub	ConcaveSurfaceProfile
:Stub	ConvexSurfaceProfile
:Stub	DatumCamberTangentialChordLength
:Stub	VaneStubDatumCamber
:StubProfile	ConcaveSurfaceProfile
:StubProfile	ConvexSurfaceProfile
:StubProfile	VaneStubDatumCamber
:VaneSub-Assembly	VaneWeldRootGap
:VaneSub-Assembly	VaneWeldConvexGrooveAngle
:VaneSub-Assembly	VaneWeldConcaveGrooveAngle
:VaneSub-Assembly	DiamondWeldPrepRootFaceThickness
:VaneSub-Assembly	DiamondWeldDepthOfPreparation
:VaneSub-Assembly	KnifeEdgeWeldDepthOfPreparation
:VaneSub-Assembly	KnifeEdgeWeldPrepThickness
:VaneSub-Assembly	DiamondWeldPrepThickness
:VaneSub-Assembly	VaneStubDatumCamber
:VaneSub-Assembly	KnifeEdgeWeldPrepRootFaceThickness
:Stub	KnifeEdgeConvexWeldPrepAngle
:Stub	AbutmentFaceLength
:Stub	VaneStubDatumCamber
:Stub	KnifeEdgeConcaveWeldPrepAngle
:Stub	StubKnifeEdgeHeight
:Stub	KnifeEdgeConvexWeldPrepAngle
:Stub	AbutmentFaceLength
:Stub	VaneStubDatumCamber
:Stub	KnifeEdgeConcaveWeldPrepAngle
:Stub	StubKnifeEdgeHeight
:Stub	KnifeEdgeConvexWeldPrepAngle
:Stub	AbutmentFaceLength
:Stub	VaneStubDatumCamber
:Stub	KnifeEdgeConcaveWeldPrepAngle
:Stub	StubKnifeEdgeHeight
:Stub	ConcaveSurfaceProfile
:Stub	ConvexSurfaceProfile
:Stub	DatumCamberTangentialChordLength
:Stub	VaneStubDatumCamber

Figure 23Identification of critical parameters of the design features

3.5.3 Case Study

This industrial case study was conducted to portray that the proposed model is able share the knowledge across the multiple product lifecycle domains. This is primarily achieved by identifying the implications of changing the design on the different domains of manufacturing. The OGV Assembly product and its related features are utilised to carry out this study, as described in Section 3.4 and in Experiment 1.

The case study identified the key parameters and their corresponding values which dictate the manufacturability of the certain feature. This was elaborated previously and shown in the below Figure 25. The below figure further shows the parameters which are critical for the manufacture of this component.

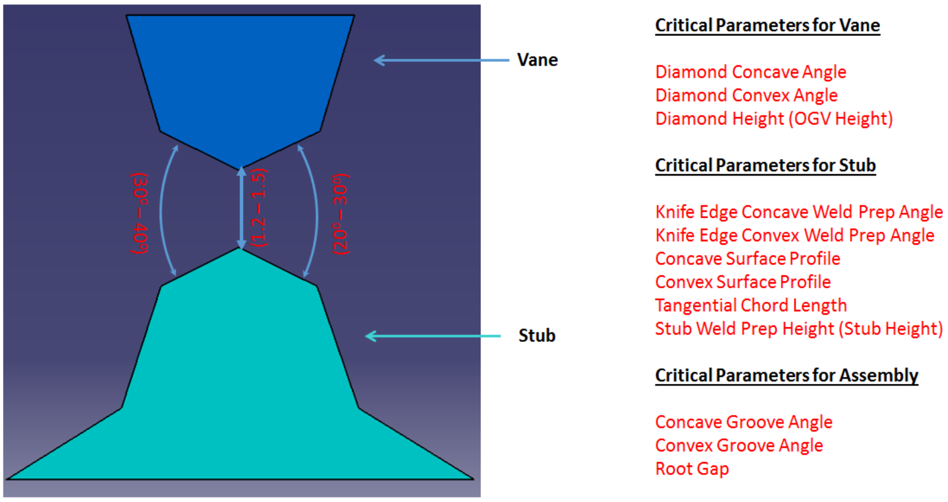


Figure 24Critical Parameters of the Design and Manufacturing Features

The different critical parameters highlighted in the above figure are for the individual components along with the assembled product. However, the parameters for the assembly are derived from those of the individual components. These derivative calculations are shown below

Concave Groove Angle = Knife Edge Concave Weld Prep Angle + Diamond Concave Angle

Convex Groove Angle = Knife Edge Convex Weld Prep Angle + Diamond Convex Angle

Root Gap = Diamond Height (OGV Height) + Stub Weld Prep Height (Stub Height)

From the case study, the limitations for the above mentioned parameters were identified as shown in the Figure 25. The values for the other parameters are shown in Table 3 below.

Table 3 Parameters and their dimensional limitations

Parameter	**Dimension**
Concave Surface Profile	0.19 – 0.38
Convex Surface Profile	0.19 – 0.38
Tangential Chord Length	250.3 – 250.9

The route to share the knowledge by identification of the overlapping or common form was already obtained and explained in Section 6.2. The following additional classes and relations that have been defined to capture the supplementary information

Table 4Additional application specific concepts/classes and relations

<table>
<tr><th colspan="2"><u>Class / Relations</u></th><th><u>Used For</u></th></tr>
<tr><td colspan="2">Parameter</td><td>To capture the different Parameters for the forms.</td></tr>
<tr><td rowspan="6">Sub Class of Parameter</td><td>Angle</td><td>To capture Groove, Weld Prep, and Diamond angles.</td></tr>
<tr><td>Height</td><td>To capture the Stub and Diamond Heights</td></tr>
<tr><td>Length</td><td>To capture the tangential chord length</td></tr>
<tr><td>Profile</td><td>To capture the Concave and Convex profile</td></tr>
<tr><td>Root Gap</td><td>To capture the Root Gap values</td></tr>
<tr><td>Thickness</td><td>To capture the Diamond and Knife Edge Weld Prep Thickness</td></tr>
<tr><td colspan="2">Manufacturable Features</td><td>To identify the features that can be manufactured</td></tr>
<tr><td colspan="2">Non-Manufacturable Features</td><td>To identify the features that cannot be manufactured</td></tr>
<tr><td colspan="2">Non-Manufacturable Parameters</td><td>To identify the parameters that cannot be manufactured</td></tr>
<tr><td colspan="2">hasDimension</td><td>To establish the relation between parameter and its sub classes with the dimension class.</td></tr>
</table>

The main aim of this case study is to further strengthen the model by identifying the consequences of changing the design parameters on its manufacturability. Vane Sub-Assembly is a critical *Design Feature* and thus its manufacturability is a vital aspect. Some of the critical parameters of this feature are the ‘Stub Height’, ‘OGV Diamond Height’, ‘Concave Surface Profile’ and ‘Convex Surface Profile ’. Therefore, a scenario where a design with the new dimensions for the above mentioned parameters are defined as indicated in the below Table 5.

Table 5Asserted dimensions for a new design

Parameter	**Dimension**
Stub Height	0.2
OGV Diamond Height	0.1
Concave Surface Profile	0.3
Convex Surface Profile	0.35

After the above dimensions were asserted into the knowledge base, the corresponding *Manufacturing Features* for the Vane Sub-Assembly was required to be identified. This was obtained by querying the knowledge base to identify the common or the overlapping forms as explained in Section 3.5.2. The following Figure 26 shows the results of the query that identifies the route to share knowledge. The highlighted portion shows the relevant path for the above mentioned feature.

DesignFeatures	CommonOrOverlappingForm	ManufacturingFeatures
StubProfile	StubProfileSurface	StubSize
Stub	StubForm	ConcaveSurface
Stub	FilletForm	EllipticalLeadingEdge
Stub	StubWeldPrepForm	StubKnifeEdgeWeldPrepHeight
Stub	StubForm	ConvexSurface
Stub	StubWeldPrepForm	StubKnifeEdgeConcaveWeldPrep
Stub	StubWeldPrepForm	StubKnifeEdgeConvexWeldPrep
Stub	FilletForm	EllipticalTrailingEdge
Stub	StubWeldPrepForm	AbutmentFace
VaneSub-Assembly	VaneStubWeld	VaneWeld

Figure 25 Identification of route to share knowledge

With, the route to share knowledge being established the manufacturing knowledge is required to be feedback for the asserted design value parameters. Therefore, a query was run to identify the *Manufacturable* and *Non-Manufacturable* features. The following Figure 27 shows the result of the query.

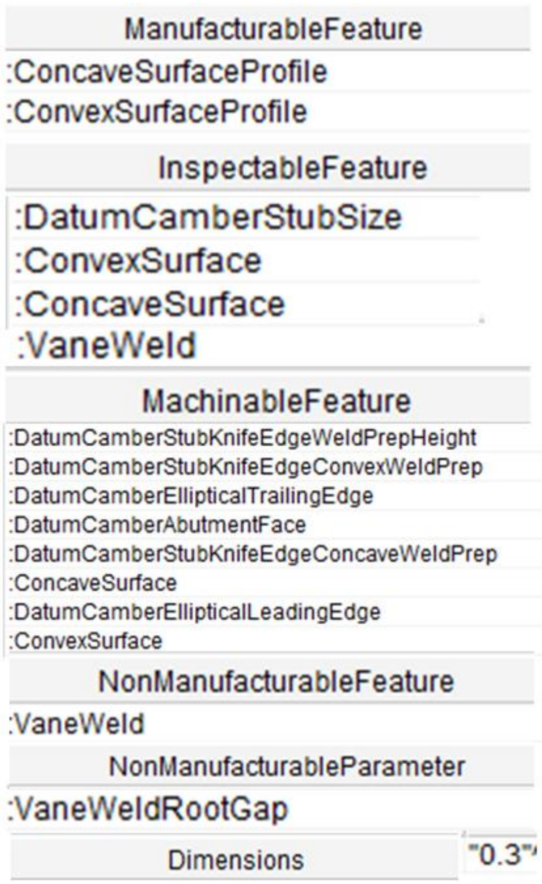


Figure 26Identification of Manufacturable, Non-Manufacturable, Machinable and Inspectable features

From the above results it can be observed that manufacturable features are machining features. However, it can be inferred that the dimensions of the ‘Root Gap’ pertaining to the welding feature is beyond the manufacturability. It implies that the asserted value for ‘Stub Height’ and ‘OGV Diamond Height’ requires to be altered. This is because these individual values dictate the dimension of ‘Root Gap’. Therefore, the values were changed as shown in the Table 6 below and asserted back into the knowledge base. The results of the querying are shown in Figure 28.

Table 6Parameters asserted with new and modified dimensions

Parameter	**Dimension**
Stub Height	1.0
OGV Diamond Height	0.3

ManufacturableFeature
:ConcaveSurfaceProfile
:ConvexSurfaceProfile
:VaneWeld

Figure 27Identification of Manufacturable features based on assertion of new dimensions

3.6 Summary

This chapter first discussed the different issues pertaining to knowledge sharing across design, machining, assembly and welding, and inspection domains. It led to defining the requirements which would enable the knowledge systems to seamlessly share knowledge across the aforementioned domains. Thus, a core set of concepts represented through a PLO was introduced. The methodology followed to develop this model has been extensively discussed in this chapter. PLO is core ontology and therefore its concepts are generic to encompass multiple product lifecycle domains. The core set of concepts and their shared relationships have been discussed in this chapter. The translational links and relations between the foundation concepts, PLO core concepts and the domain concepts have been elaborated as well.

PLO is a core ontology as it has been constructed to be an intermediate layer and not a pure design or manufacturing domain ontology. To understand this, each core concepts have been thoroughly explained informally followed by the combined illustration of PLO. The informal description of the core concepts have been formalised using OWL. The formalisation process has been explained through the use of different namespaces, classes, properties, restrictions, rule and axioms.

The route for knowledge sharing is elaborated, i.e the framework that promotes the knowledge from machining, assembly, welding and inspection to be shared with design. It must be noted that in this chapter the knowledge sharing aspect is elaborated from a high level perspective. That is, the share-ability of the combined knowledge from multiple domains to design is explored here.

And lastly, the semantic capture of the PLO concepts is experimentally verified. This encompassed the verification of the specialisation levels in capturing the varying depths of meanings of concepts. PLO was then validated for its applicability as semantic base between the domains of design, machining, welding and inspection. This was carried out via testing its ability to develop semantically rigorous application specific ontologies. Together with this, PLO was utilised to provide a route to share the knowledge between the application specific ontologies. The elaboration of the assembly and welding specific knowledge has been described in Chapter 4. The manufacturing operation related knowledge sharing is further discussed in Chapter 5.

4. PLO as a Semantically Enriched Welding Ontology

This chapter elaborates the utilisation of PLO as a welding ontology to achieve interoperability. It is primarily used to capture the welding knowledge and consolidate the welding standards. In this chapter, the following contributions have been discussed

- The semantic inconsistency issues in welding standards are firstly investigated systematically.
- A solution model is proposed to capture the semantics of the welding concepts.
- The semantics of the core concepts of the model are further adapted as per the definitional requirements of the welding specific standards to resolve the semantic issues within and across them.

This chapter is organised as follows: Section 4.1 describes the standardisation within the welding domain, as well as the requirements for using an ontological approach in representing the standards for interoperability. Section 4.2 portrays the proposed semantically enriched model to capture the welding knowledge. Section 4.3 explains the formalisation of the proposed ontology. Section 4.4 explains the consolidation of the welding standards. The model is experimentally verified in Section 4.5. And, finally Section 4.6 summarises this chapter.

4.1 Limitations of Welding Standards and Requirements to Consolidate

The potential utilisation of generic standards to overcome interoperability problems have been discussed in Chapter 2. Similarly there are standards which have been established for interoperability across welding domains. This section reports on the breadth of welding standards before investigation into their semantic inconsistency and interoperability issues.

4.1.1 Standardisation for Welding

Various standards have been developed to support interoperability between welding and design domains. Standards also attempt to regularise welding processes as there are multiple categories based on material conditions and applications. The International Organisation for Standardisation (ISO) community has developed the ISO/TR 25901 standards as agreed global references for welding. Although the ISO is global in scope, there are various national organisations and committees that have developed their own standards to meet local industrial requirements. For example, the American Welding Society (AWS) has developed its own standards for American industries, while the British Standard Institution (BSI) has done the same for the UK.

The scope of the ISO welding committee is for "*Standardisation of welding, by all processes, as well as allied processes; these standards include terminology, definitions and the symbolic representation of welds on drawings, apparatus and equipment for welding, raw materials (gas, parent and filler metals) welding processes and rules, methods of test and control, calculations and design of welded assemblies, welders' qualifications, as well as safety and health.*" (ISO/TC44, 2017). It signifies that the committee looks after all the regularisations as well as unveiling of the best practices within the welding domain.

Even though ISO defines the international welding standard ISO/TR 25901, there are several other regional standards developed by different welding communities. Table 7, shows the major standardisation bodies involved in development of welding standards along with the corresponding technical committees involved. Manufacturing companies prefer a multi-standard based approach to address various industrial requirements. However, the considerably large number of standards available poses a problem for interoperability owing to a lack of compatibility of the terms used. This is primarily because they are defined in different ways even though their practical use could be the same or similar. The focus of this part of the research is on the accurate capture of the semantics of the core terms used in the standards and overcome their semantic inconsistency.

Table 7Major welding standardisation bodies and standards

Organisation	Level of Authority	Jurisdiction	Welding Technical Committee (TC)	Sub Committee for Vocabulary (SC)	Standard Name
International Standard Organisation (ISO)	International	Worldwide	ISO/TC 44	ISO/TC/44 SC7	ISO/TR 25901-1:2016, ISO/TR 25901-2:2016, ISO/TR 25901-3:2016, ISO 25239-1:2011
European Committee for Standardisation (CEN)	Continental	European Union	CEN/TC 121		PD CEN/TR 14599:2005
American Welding Society (AWS)	National	America	AWS A2	AWS A2B	AWS A3.0M/A3.0:2010
British Standards Institution (BSI)	National	United Kingdom	BS WEE/1		BS 499-1:2009

4.1.2 Semantic Inconsistency in Welding Standards

For the welding standards reported earlier, there are various semantic inconsistency issues (Saha et al., 2017), which are required to be resolved to support interoperability. This research will focus on text based semantic inconsistency, and the investigations are from two perspectives:

1. Inconsistencies within the same standards, and
2. Inconsistencies across different standards.

The terms and definitions in the welding standards were found to be highly textual, making them open to human interpretation and therefore inefficient and error-prone for interoperability. The textual nature and the subjective interpretation of the standards are corroborated by the following example:

Welding is defined in AWS as *"A joining process that produces coalescence of materials by heating them to the Welding Temperature, with or without the application of pressure or by the application of pressure alone, and with or without the use of filler metal"* (AWS A3.0).

This definition introduces the term *Welding Temperature*, which is not explained in the standard and open to any interpretation. *Welding Temperature* can be perceived as any temperature. This raises ambiguity or misinterpretation of the various welding processes as it relies on every individual's perception on this term. This will further affect the categorisation of welding processes. The commonly believed understanding of *Welding Temperature* is that it is the melting temperature of the substrate material. However, agreeing on this definition the categorisation of the *Solid State Process* as a welding process is semantically inappropriate.

The *Solid State Process* is defined in AWS as *"A group of welding processes that produce coalescence by the application of pressure without melting any of the joint components."* (AWS A3.0)

According to the definition, there is no melting involved in the process. This is a contradiction of its enlistment as a welding process. Therefore, without the clarity on the semantics of *welding temperature*, there will be confusion with regards to categorisation of other welding and joining processes. Therefore, it is essential to capture the semantics of welding temperature for each specific welding process. On the other hand the *Fusion Welding Process* is defined in AWS as "*"Any welding process that uses fusion of the base metal to make the weld."*(AWS A3.0).

Another example is from the ISO welding standard. The standard does not use the term *Welding Temperature* but it shows ambiguity in terms of interpretation as well. For example, the definition of *Welding* in the ISO standard is: *"Joining process in which two or more parts are united producing a continuity in the nature of the workpiece material(s) by means of heat or pressure or both, and with or without the use of filler material"* (ISO – TR 25901-1) (CEN – TR 14599 [EN 1792]) (BS 499-1).

This definition does not describe the condition of the substrate material during the process. Hence, its interpretation might lead to erroneous classification of not only the welding processes but also the other joining processes. Depending on the standard implemented for industries, it will recognise and interpret the semantics for that particular standard. Hence, systems implemented with AWS and ISO semantics will struggle to share knowledge with each other.

In the following, text based semantic inconsistencies within the same standards and across different standards will be further elaborated.

4.1.2.1 Inconsistencies within the same standard

On investigation, inconsistencies were found within the same standards. For example, in the AWS A3.0 standard, the categorisation of the *Resistance Spot Welding* and *Resistance Seam Welding* processes was found to have issues. The standard classifies both of the processes as *Fusion Welding* as well as *Solid*

State Welding, which is a violation of the fundamental semantics of their definition as defined previously. Similarly, *Fusion Welding* has been defined in the ISO standard as *"Welding without application of external force in which the faying surface(s) has (have) to be molten; usually, but not necessarily, molten filler metal is added"* (ISO /TR 25901). And *Welding with Pressure* is defined as *"Welding in which sufficient outer force is applied to cause more or less plastic deformation of both the faying surfaces, generally without the addition of filler metal"* (ISO /TR 25901). The key attribute that differentiates the two processes is the condition of the substrate material during the joining process itself. This fundamental difference prevents the categorisation of the same process in two different categories as it violates the inherent semantic rationale of the definitions.

Moreover, in both AWS and ISO standards, *Braze Welding* has been classified as a *Brazing* process. However, the process does not involve any capillary action, which is a fundamental requirement for *Brazing*. Furthermore, there is no melting of the substrate material thereby casting a doubt over the process being termed as *Welding*. The interpreted semantics of the processes might classify it as a *Solid State Welding* process as well. Hence, this categorisation is debatable and inappropriate for interoperability due to the prevalent inconsistencies.

4.1.2.2 Inconsistencies across different standards

Similar shortcomings were found across multiple standards. For example, the ISO standards denote *Solid State Welding* as *Welding with Pressure*. It is entitled to encompass all the processes where the coalescence occurs due to pressure. However, some of the processes which have been classified within this category are also categorised as the *Fusion Welding* process in the AWS standard. The definitions of the processes in ISO have been explained previously and those in AWS are depicted in Table 8. Although the definitions are not entirely identical, the overarching theme of them is similar. The varying categorisation across multiple standards is depicted in Table 9. This portrays the violation of semantics of the definitions across them. The standards are mutually incoherent, further compelling customers to follow any particular standard in a multinational environment.

Table 8 Definition of Joining Processes in AWS Standard

Fusion Welding (AWS A3.0) *"Any welding process that uses fusion of the base metal to make the weld."*
Solid State Welding (AWS A3.0) *"A group of welding processes that produce coalescence by the application of pressure without melting any of the joint components."*

Table 9 Inconsistencies across different standards

Process	**ISO**		**AWS**	
	Fusion Welding	*Welding with Pressure*	*Fusion Welding*	*Solid State Welding*
Percussion		X	X	
Projection		X	X	
Flash		X	X	
Resistance Spot		X	X	X
Resistance Seam		X	X	X

The investigation has highlighted the issues faced for welding interoperability using these standards. It was understood that some inconsistencies in the standards are self-contradictory, some are categorised wrongly and some are mutually non-reconcilable in their current form. This makes it more evident that it is imperative to have a more rigorous, consistent and computer interpretable categorisation and definition of welding concepts.

These concepts are, however, required to be defined more rigorously at a generic level for all types of joining processes that can further constrict to welding concepts. This is achieved through PLO with a slight extension. The need to capture knowledge at different levels of abstraction enforces the need to have concept definition from a generic to specific welding level. The principles of the specialisation levels have already been discussed in Section 3.2.1.

4.2 A New Approach towards Defining Semantically Enriched Welding Concepts

In order to overcome the above mentioned issues the proposed PLO has been utilised with some extensions and modifications. The lightweight structure of the proposed model is shown in Figure 30. This model is developed to establish a foundation for consolidating welding processes that resolves semantic inconsistency. Furthermore, the model acts as a base to consolidate the welding standards. It must be noted that these concepts have been extended from the PLO model in Figure 10 through the concepts *Process*, *Manufacturing Process* etc.

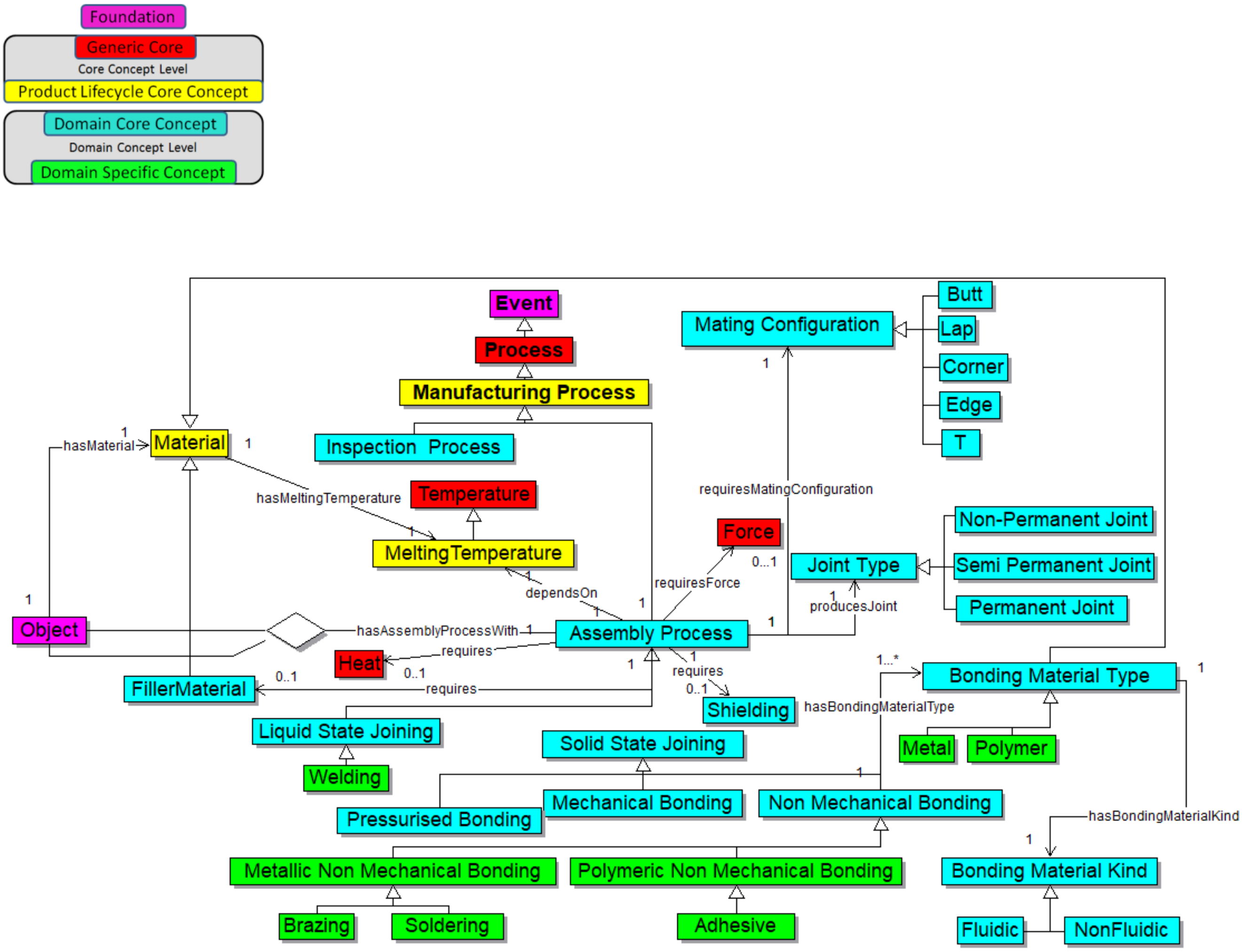


Figure 28PLO Extension Model for Welding Knowledge Capture

The crucial aspects of modelling are explained below:

1. The range of key concepts are defined with semantics generic enough to provide a base for any joining process and also specialised enough to provide a direct route for aligning with specific welding standards. For example, in Figure 29, *Assembly Process, Liquid State Joining*, are the key concepts identified for joining in order to build up the ontology. The concept of *Welding* is also identified to align the welding standards.
2. Based on the key concepts, relationships between them are specified, e.g., the relationship *hasAssemblyProcessWith* is defined between two *Object* using Ternary Operator.
3. The model is devoid of any inconsistencies and incoherencies as it is developed by explicitly defining the semantics of the concept. For example, *Metallic Non Mechanical Bonding Process* and *Polymeric Non Mechanical Bonding Process* are defined as two different sub processes of *Non Mechanical Bonding Process*. This ensures the consistent capture and categorisation of non-mechanical bonding processes with subtle semantic differences.

The modelling process of this structure is similar to that carried out in Section 3.2.3. This includes the explicit definition of the core and the domain concepts specific for this section of the model. The definitions and the meanings of the terms related to joining and welding found in the literature along with the widely accepted standards have been studied. Some of the definitions have been adopted from the published literature and the standards while the others have been newly proposed. The essential classes have been defined and explained in the following paragraphs.

The *Manufacturing Process* class is a generic concept for the entire product lifecycle including design and manufacturing and has been defined as *"Structured set of activities or operations that is performed upon an object and contributes towards converting it from a raw material or a semi-finished state to a state of further completion"*. *Assembly Process* class is a specialisation of this class which has been defined as "*Process by which a group of components are brought together under specific mating configuration to form a unit"*. It includes all those processes in which two or more components are joined together through some form of Bond. Depending on the type of the Bond there can be different *Joint Types*. These are *Permanent Joints, Non-Permanent Joints* and *Semi-Permanent Joints* and are based on the condition of the mating component on bond removal. The *Mating Configuration* class describes the orientation in which every joint is aligned. The different configurations in which they are sub classified are *Butt, Lap, Corner, Edge* and *T.*

The key factor which differentiates the processes was found to be the joint type as well as the fundamental process by which the bond is created. Another crucial differentiating criterion is the procedure by which the bonding takes place. It also depends on the state of the mating component material which can either be liquid or solid. This can be conceived as the primary basis for the classification of the *Assembly Processes* and thus on a holistic level they are classified as *Liquid State Joining*, defined as *"Process that categorises all the joining processes in which the participating component(s)/substrate material passes through the liquid state before forming the joint"* and *Solid State Joining* processes are described as *"Process that categorises all the joining processes in which the participating component(s)/substrate material remains in the solid state throughout the process of forming the joint".*

From the definitions, *Liquid State Joining* process was found to be the most relevant category for subsuming *Welding* class. The justification of such a classification can be found in the origins of the word *Welding*. It is an "Alteration of 'well' and in the obsolete sense means 'melt or weld' heated metal (late 16th century)" (Oxford, 2010). Agreeing on the origins it could be understood that the welding essentially refers to processes where there is an involvement of actual melting of the metal. Fundamentally this means that mating components partly go through the liquid state during the joint forming process. Hence, accordingly *Welding* process should encompass all those processes where there is some form of melting of the metal. From this perspective all the fusion processes should be classified as welding processes.
Solid State Joining class is a contrast to *Liquid State Joining* and has further categorization. It is further categorized as

1. *Non Mechanical Bonding Process* which has been defined as "*Solid State Joining Process where the bond or the joint between the mating objects interface are produced by a fluidic substance. The fluidic substance acts as the bonding element or material."*
2. *Mechanical Bonding Process* defined as *"Solid State Processes where the bond or the joint between the mating objects interface are produced by a Non-Fluidic or Solid Material such as a Fastener which is used for joining securely and temporarily."*

3. *Pressurised Bonding Process* which has been defined as *"Solid State Joining Process where the bond/joint is primarily produced by application of any form of pressure with or without any filler metal."*

Non Mechanical Bonding Process is further broadly classified as *Metallic Non Mechanical Bonding Process* and *Polymeric Bonding Process*. The differentiating attribute for the two classes of processes lies on the nature of the fluidic material that is used for bonding which can either be *Metal* or *Polymer*. The proper definition of the concepts is an essential step to capture their semantics and further highlight the additional requirements for formalisation. The inter class relationships are also revealed through their proper description.

From the UML model in Figure 30, it can be observed that all the relationships are defined at a generic level for the super classes. This is based on the understanding that the sub classes inherit all the attributes of their super classes which includes all of their relationships. The prefix of the relationships denotes the specialisation level they exist in and further their dominion. For example, the prefix *Core Domain* denotes that the concerned relation is between the core domain concepts while *Multi* defines inter-level relationships. Within the hierarchical model the primary relationships originate at *Assembly Process* class. Different cardinality has been assigned for the relationships depending on their constraining requirements. The model in Figure 30 describes the generic level the concepts and the complex relationships between them. The different relations along with the corresponding classes they connect are shown in Table 10. These are required to describe an assembly process as well as its further categorisation.

Table 10 Relations and classes they connect

Relations/Properties	**Domain**	**Range**
CoreDomain:requiresMatingConfiguration	*AssemblyProcess*	*MatingConfiguration*
CoreDomain:producesJoint	*AssemblyProcess*	*Join Type*
CoreDomain:requiresShielding	*AssemblyProcess*	*Shielding*
CoreDomain:hasBondingMaterialType	*NonMechanicalBonding*	*BondingMaterialType*
Multi:requiresForce	*AssemblyProcess*	*Force*

Multi:requiresHeat	*AssemblyProcess*	*Heat*
Multi:dependsOn	*AssemblyProcess*	*MeltingTemperature*
CoreDomain:requiresFillerMaterial	*AssemblyProcess*	*FillerMaterial*
CoreDomain:hasBondingMaterialType	*BondingMaterialType*	*BondingMaterialKind*

4.3 Consolidation of the Standards and Formalisation of the Model

The UML model that consolidates the welding standards from PLO is shown in Figure 30. It portrays the utilisation of the core concept *Welding* as a base that provides an avenue to consolidate the welding standards. The entities are captured in the form of classes and relationships. The core concept *Welding* has been defined at the generic level to provide the very basic level of semantics to consolidate the welding standards. A specialised relation *requiresMaterial* has been defined for the *Welding* class as this is a specific requirement for this concept. The definitions of welding found in different standards are denoted as specialised classes such as *AWS:Welding*, *ISO:Welding* etc. These definitions are tailored to the core concept *Welding* through the subsumption relation. *Multi:requiresParts* and *Multi:requiresWeldingTemperature* are the two relationships which are defined specifically for capturing the semantics of the definitions found in the two set of standards.

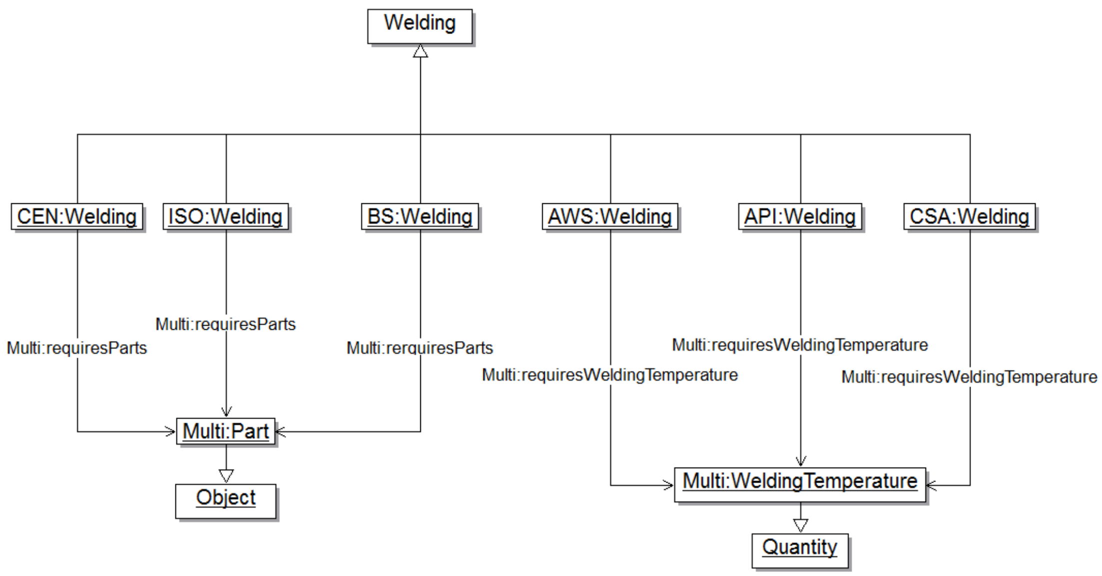


Figure 29Ontology model for consolidation of welding standards

4.4 Implementation and Formalisation of the Model

This section elaborates the implementation procedure of the model. The overall high level implementation framework has been explained in Section 3.2.3. Figure 31 specifically elaborates the Step 3 of the framework in Figure 2. The concept *Welding* is used to explain the implementation of the proposed framework to capture the welding knowledge and for the consolidation of the standards.

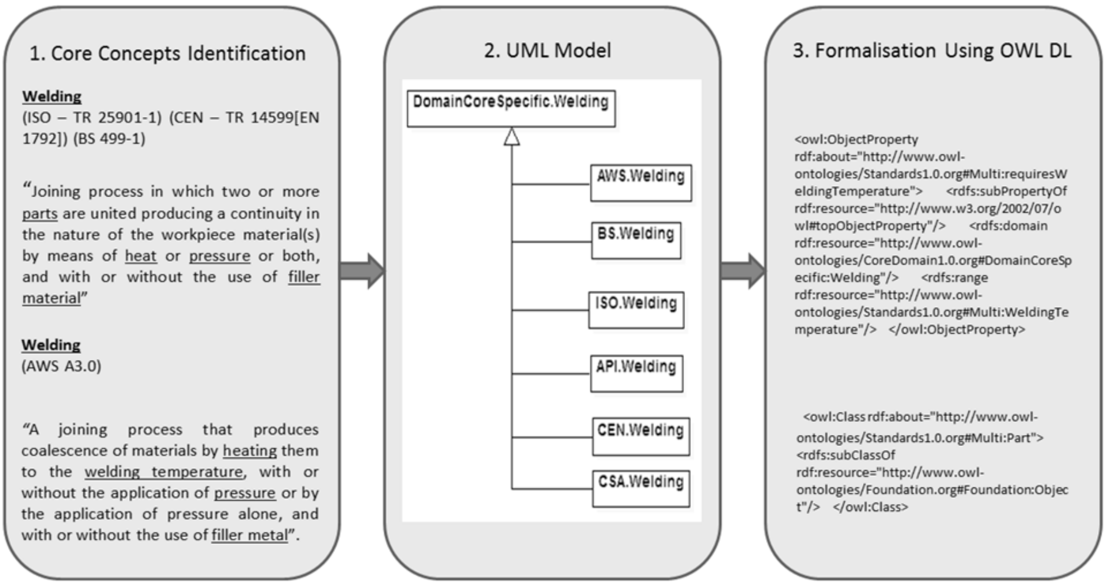


Figure 30Implementation of the framework

The first step in the implementation of the framework was to identify the core concepts from the natural language definition of the concepts. This requirement was achieved through a survey of all the welding standards. The compilation of similar terms and their informal definitions revealed the important key words and their respective sentences where they are used. The term *Welding* was found to be referenced across multiple standards. Hence, it is used as a core concept to which the definitions from different standards are tailored. This step also involved a crucial input from the domain expert from the industry as it helped in identifying the other important concepts which are cross referenced across different definitions.

Based on the identified core concepts and their relationship, the UML model was constructed. The keywords highlighted within the textual definitions in this step identify the key set of concepts along with their relationships in the model. In the final step, the model is formalised into a consolidated ontology using the description logic based language OWL DL. The formalisation procedure of this model is similar to that explained in Section 3.3.3 as it essentially is an extension from PLO. The final step of the implementation process is the experimental verification of the model.

4.5 Experimental Validation

The successful capture of the welding knowledge through PLO and its extension is described in this section. Furthermore, the models capability to consolidate the welding standards has been verified. This section describes the various test cases carried out on the ontology to verify the capture of its requirements earlier. This includes the following:

1. Consistency checking of the formalised ontology,
2. Verification of semantic capture
3. Inference of new taxonomy and Consolidation of the welding standards.

The inconsistencies and the limitations of the text based semantics are shown through the formalisation of the *Welding* concept. This is followed by assertion of 'Friction Stir Welding' process.

1. The AWS standard requires *Welding Temperature* to be equal to the base material's *Melting Temperature* for the formalised definition of *Welding* process. However, the Friction Stir Welding process does not involve any melting of the base material implying that the welding temperature is lower than the melting temperature. The standards classify it as a Welding process. Thus, assertion of Friction Stir Process results in inconsistency which the system interprets and displays the following error message including its explanation in Figure 32.

   An Error Occurred During Reasoning

   Inconsistent Ontology Exception: Cannot do reasoning with inconsistent ontologies!

   Reason For Inconsistency: Individual http://www.owl-ontologie/PLO.org#SteelMeltingTemperature is forced to belong to class http://www.owl-ontologies/PLO.org#BaseMaterialMeltingTemperature and its complement

   Figure 31 Error and explanation due to assertion of inconsistent semantics

2. The taxonomy of all the concepts starting from the foundation level to the specific domain of Welding is illustrated in Figure 33.

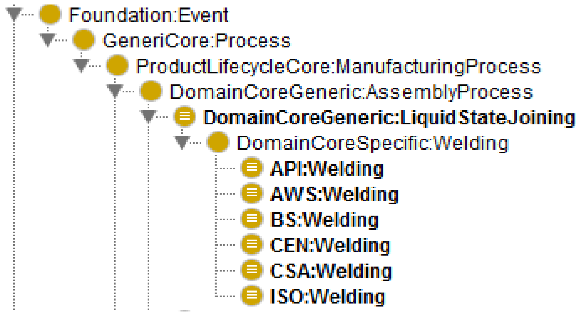


Figure 32Taxonomy of classes/concepts

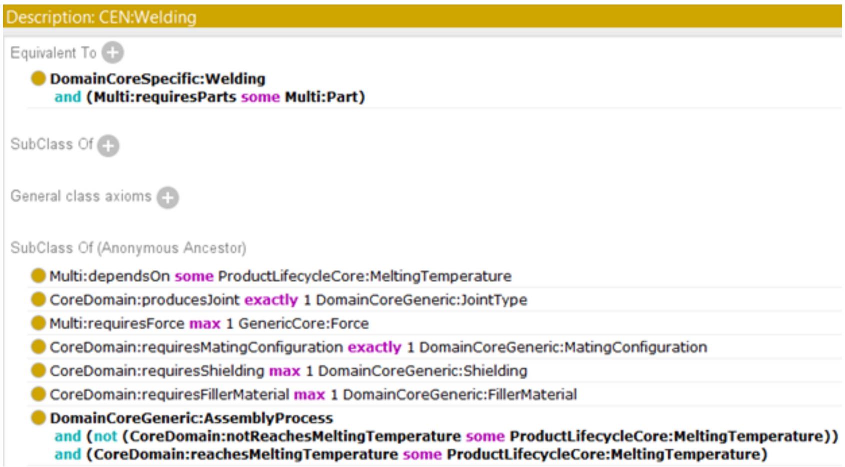


Figure 33Restrictions on the classes/concepts

The capture of the ‘necessary conditions’ for the *Welding* shown in Figure 54 is through the properties defined within ‘SubClass Of’ category. These semantics are reused along with several specifics for each specialisation levels. They contribute towards distinguished definitions of joining processes. Figure 34 also shows the capture of the ‘necessary and sufficient conditions’ through the ‘Equivalent To’ category. The inherited as well specific properties assigned for the *CEN:Welding* concept is shown. These conditional properties expedite consistency checking of the instantiated information. Figure 35 shows the assertions of the instance, ‘Friction Stir Welding’ of *Welding* class. The missing semantics are marked with red which prompts the user to populate these values. The Pellet reasoner has been used to check the consistency of the ontology which includes assertions of instances. This results in error as shown in Figure 36 along with its explanation. The inherent semantics of the class defined through the assigned properties results in this error. This verifies the capture of semantics of the proposed model and the corresponding definitions of the concepts.

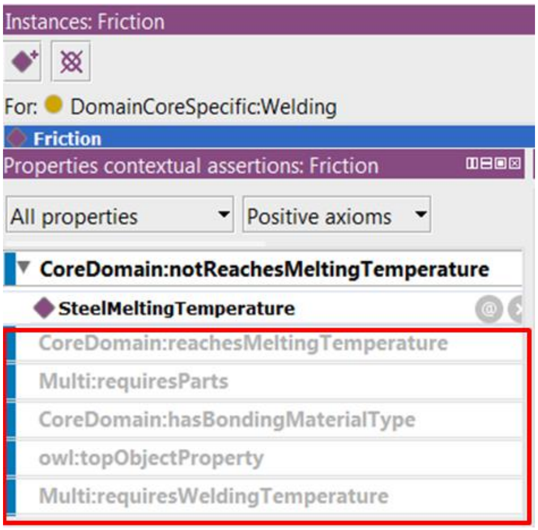
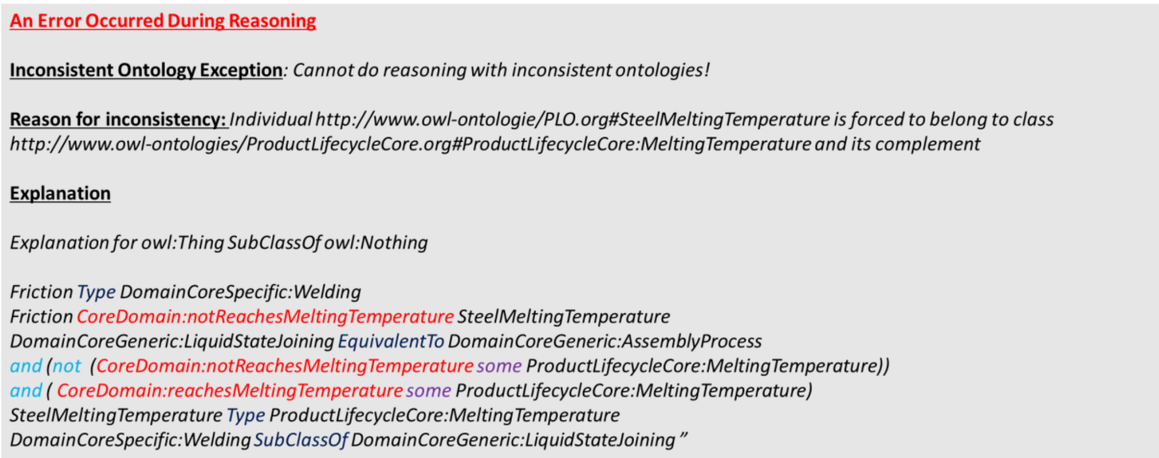


Figure 34Identification of missing semantics for asserted class



An Error Occurred During Reasoning

Inconsistent Ontology Exception: *Cannot do reasoning with inconsistent ontologies!*

Reason for inconsistency: *Individual http://www.owl-ontologie/PLO.org#SteelMeltingTemperature is forced to belong to class http://www.owl-ontologies/ProductLifecycleCore.org#ProductLifecycleCore:MeltingTemperature and its complement*

Explanation

Explanation for owl:Thing SubClassOf owl:Nothing

Friction Type DomainCoreSpecific:Welding
Friction CoreDomain:notReachesMeltingTemperature SteelMeltingTemperature
DomainCoreGeneric:LiquidStateJoining EquivalentTo DomainCoreGeneric:AssemblyProcess
and (not (CoreDomain:notReachesMeltingTemperature some ProductLifecycleCore:MeltingTemperature))
and (CoreDomain:reachesMeltingTemperature some ProductLifecycleCore:MeltingTemperature)
SteelMeltingTemperature Type ProductLifecycleCore:MeltingTemperature
DomainCoreSpecific:Welding SubClassOf DomainCoreGeneric:LiquidStateJoining "

Figure 35Error generation due to assertion of instances with inconsistent semantics

3. The restrictions implemented into the model allow inference of new knowledge. The inferences are normally deductive, inductive, abductive or analogical (Farhad Ameri, 2015). The inferred hierarchy of the classes as shown in Figure 37 was obtained after the ontology was reasoned. This allowed the identification of commonalities between the different *Welding* classes as it revealed the subsumptions and equivalency. The reasoner is able to identify the equivalency between two groups of classes as shown in Figure 37. One of the group comprised of *AWS:Welding*, *CSA:Welding* and *API:Welding* classes. The other group was that of *ISO:Welding*, *CEN:Welding* and *BS:Welding*. This verifies that the proposed model is able to consolidate the welding standards through the definition of welding as stated in different

standards. Therefore, the model is verified to provide tailored semantics for welding standards that remove the highlighted issues, connects them, makes them consistent and provides a base for interoperability across them.

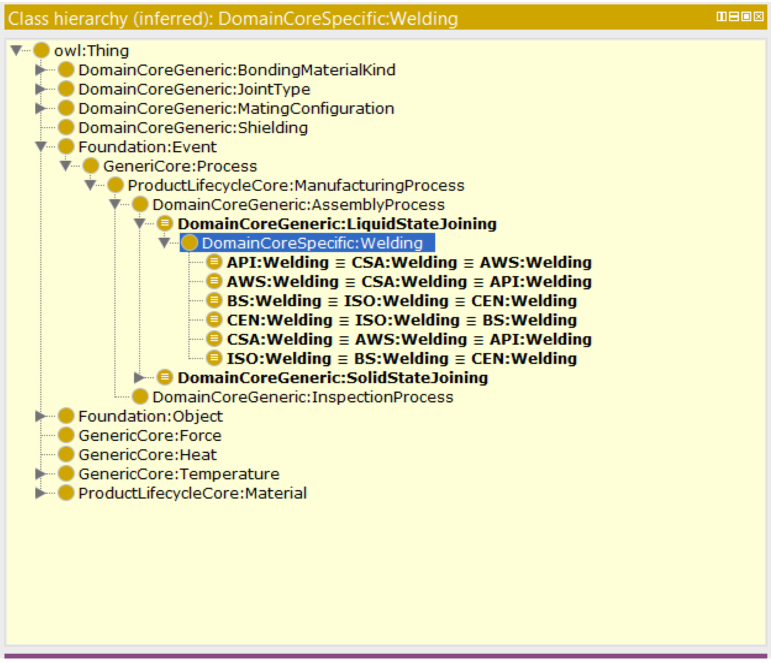


Figure 36Inference of new taxonomy and consolidation of the standards

The extent of interoperability is further enhanced using the SWRL rules. The enhanced capability is verified by querying the knowledge base to interpret and infer the processes which are similar but have been differentially termed across the standards. The following Figure 38 shows the query and the result. Based on the defined rules, the system is able to identify that the 'Gas Tungsten Arc Welding (GTAW)' of the AWS standard is same as the 'Tungsten Inert Gas (TIG)' welding process of the ISO standard. This verifies the capability of PLO to capture, share the welding knowledge and consolidate the welding standards.

SQWRL Query:

isEquivalentTo(?AWS,?ISO) -> sqwrl:select(?AWS,?ISO)

Result:

AWS	*ISO*
:GTAW	*:TIG*

Figure 37Identification the similar processes referred differentially across the standards

4.6 Summary

This chapter has elaborated the utilisation of PLO to capture the welding knowledge and further consolidate its related standards. The semantic inconsistency issues in welding standards are systematically investigated first. An extension of PLO is proposed to formally capture the semantics of welding concepts. The proposed model is utilised to resolve the semantic inconsistencies within and across welding standards. Furthermore, the model is utilised to facilitate the knowledge sharing across welding domains that use different standards through their consolidation. And lastly, the capability of PLO to capture the welding knowledge has been experimentally verified along with the consolidation of the standards.

5. PLO as a Manufacturing Ontology To Capture Manufacturing Operations and Sequencing Knowledge

This chapter showcases the development of PLO to capture the knowledge pertaining to manufacturing operations and their sequencing. In this chapter, the following contributions have been elaborated

- Different kinds of manufacturing operations and their probable sequential orders prevailing within the manufacturing world are discussed.
- A solution through PLO and its extension is proposed to identify and categorise these manufacturing operations.
- The capability of the proposed ontology transcending to infer the sequences of operations was shown. Furthermore, OWL was exploited for the formalisation of the proposed ontology.

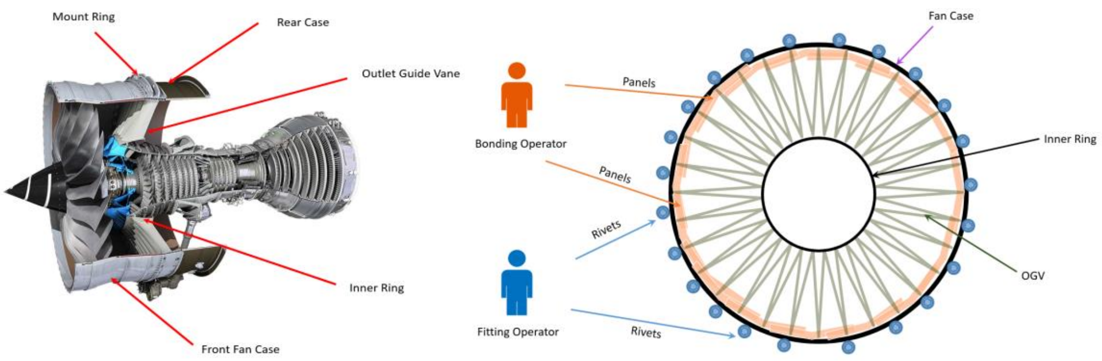
The chapter is organised as: Section 5.1 describes the requirements and complexity involved in modelling manufacturing knowledge. Section 5.2 portrays the proposed semantically enriched ontological model highlighting various scenarios of operational sequences in a manufacturing environment. Section 5.3 explains the formalisation of the proposed ontology. The proposed solution was experimentally verified in Section 5.4. And, finally Section 5.5 summarises this chapter.

5.1 Requirements to Model Manufacturing Processes and Operation Sequencing Knowledge

This section elaborates the manufacturing knowledge required for modelling process planning activities. Table 11 partially shows a typical process plan for manufacturing an aero engine fan case illustrated in Figure 39. The creation of such a fan case encompasses various manufacturing operations, such as machining, assembly, welding and inspection. The fan case is generally comprised of three modules: Front Case, Rear Case and Outlet Guide Vanes (OGV) assembly. In Table 11, the process commences with a turning operation on an “Inner Ring”, followed by a machining (milling) operation to create the vane attachment feature required for the next operation. After the “Inner Ring” is machined and inspected, the “OGV” is welded onto the “Inner Ring” vane attachment feature via two different operations: tack and final weld. This is followed by assembly operations where the “Mount Ring” is bolted with the “OGV” and fitted to the “Front and Rear Cases” via bolting features. The assembly also encompasses the fitment of other nuts and rivets. After the mechanical fitment, different panels are bonded onto the interior of the fan case using panel bonding fitment features.

Therefore, to ensure that the product is made to specification, a process planning activity is required to enable the above the mentioned operations are carried out in a correct manner (sequencing). The process plan primarily contains the following crucial information:

1. Details of every feature being created in each operation and the possible sequences of operations for manufacturing a product;
2. Specifications of the tools, fixtures and machines for each operation;
3. The technical instructions describing every activity for operators to carry out each operation.



(a) An aero engine and its fan case (b) The front view of the fan case assembly

Figure 38An aero engine and a fan case

Table 11Operations for process planning

Operation	Description	Machine	Fixture	Tool
OP 10	Turning	WandB CNC W-8476587	Turning Fixture TF-847365	DNMG 150 Grade 8, Insert RCMT 10T3MX-F1
OP 20	Machining	5 Axis Machine Centre 3 W-438765	Internal Milling Fixture MF-985789	T105, T32, T722, T316
OP 60	Machining	Machine Centre 1 W – 3456	Milling Fixture MF-585789	T115, T22, T122, T116

OP 50	Inspection	CMM Large W-489765	Inter Shop Pallet T1 I/Ring, Lifting Frame T1 Inner Ring	RAD 12 Gauge, , RAD Gauge Set, Ultrasonic Gauge
OP 30	Tack Weld	Circum Bode	Vane Weld Fixture	Welding Equipment, Argon Backer, Argon Box
OP 40	Hand Weld	Circum Bode	Vane Weld Fixture	Welding Equipment, Argon Backer, Argon Box, Welding Filler, Mandrel
OP 80	Fitting	Fitting Bay	Fan Case Assembly Fixture AF-456890	Torque Wrench, Screws, Rivets, Brackets
OP 60	Bonding	Bonding Oven	Fan Case Assembly Fixture AF-456890	Panel Fillers, Linner Fillers

One of the critical decision-making attribute within process planning is to choose a suitable sequence of operations from various possibilities. As illustrated in Figure 40, generally, every operation can be categorised as:

1. Autonomous,
2. Semi-Autonomous, or
3. Manual

Section 5.3 provides a detailed explanation of the categories along with their classification procedures. In a manufacturing environment, these types of operations can precede each other as shown in Figure 40. The different types of operations are based on the example described in Table 11 and annotated in the legend.

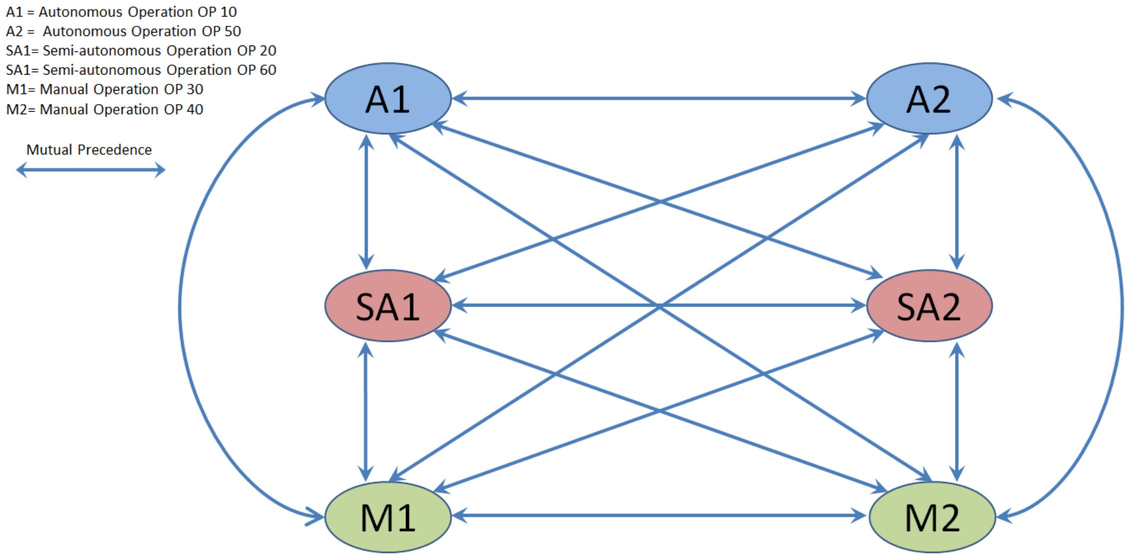


Figure 39 Various sequences of operation

The decision for the precedence of an operation to another depends on various factors, such as:

1. The types of machines (e.g. machining centres, milling machines) and fixtures involved;
2. The state of resource tools (e.g. cutting tools) and their orientations;
3. The constraints from the geometry and orientation of a part.

The industrial study identified that not all the operations are necessarily carried out in a sequential order for assembly. Some of the operations can be performed in conjunction with each other. For example, in the fan case assembly process, the preparation and fitting of the panels can be carried out concurrently with the riveting operation although they are different operations in the process plan shown previously. Figure 39(b) illustrates this, as it shows that “Fitting Operators” carry out the riveting operation on the exterior of the fan case. At the same time, the “Bonding Operators” are involved in the fitment of the panels on the internal face of the fan case. This scenario enhances the complexity to model process planning activities. The relevant information should be represented in an ontology model to support the process planning and sequencing processes. The next section further elaborates on this.

5.2 Utilising PLO to Capture the Manufacturing Operations and its Sequencing Knowledge

5.2.1 Modelling the Ontology, Defining the Concepts and Relations

The modelling procedure for PLO has been discussed already in Section 3.2.3. The extension of PLO through addition of more classes and subclasses follow the same methodology. Therefore, the developed model is shown in Figure 41. It must be noted that these concepts have been extended from the PLO in Figure 10 through the concepts *Operation*, *Manufacturing Operation* etc.

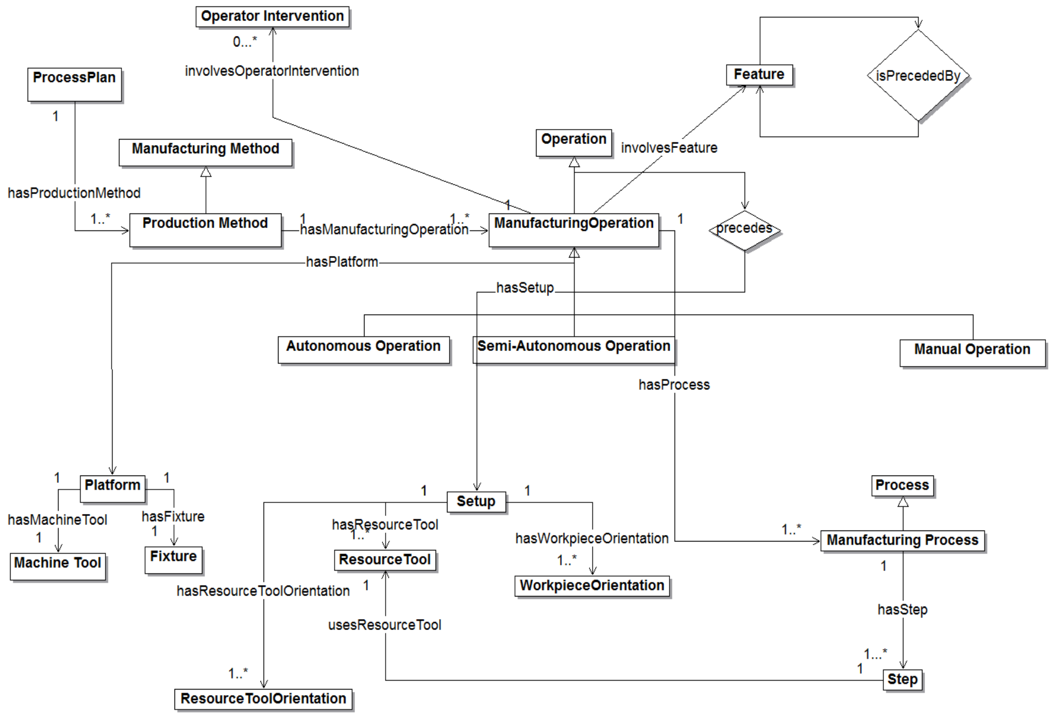


Figure 40PLO Extension Model for Manufacturing Operations

The proposed definitions for the top level concepts in Figure 41 are given in Table 12. The different relationships defined between the concepts are explained in Table 13.

Table 12Core concepts and definitions

Operation "Activity or group of activities in which an entity is altered/manipulated. It uses a particular Setup."
Manufacturing Operation "Activity involving but not limited to, the machining, welding, heat treating or other processes utilized to produce a finished product."
Autonomous Operation "Operations in which there is no external human intervention or disruption unless there is an emergency."
Semi-Autonomous Operation "Operations which are performed partially in an autonomous state and require operator intervention only to manually change the orientation of the tool/workpiece, adjusting the tool etc."
Manual Operation "Operation in which the operator performs the entire operation manually and is totally under human control."
Process "An event or series of events resulting in a change of state"
Manufacturing Process "Structured set of activities or operations that is performed upon an object and contributes towards converting it from a raw material or a semi-finished state to a state of further completion."
Method "Process by which a task is completed"
Manufacturing Method " A sequence of events involved in the manufacture of a component"
Production Method "Process by which the task of manufacturing a product is completed"
Platform "It is a combination of Machine Tool and Fixture for a particular operation."
Setup "An event with a specific Resource Tool, Resource Tool Orientation and Workpiece Orientation"

Table 13Relationships with their domains and ranges

Relations/Properties	**Domain**	**Range**
hasFixture	*Platform*	*Fixture*
hasMachineTool	*Platform*	*MachineTool*
hasManufacturingOperation	*ProductionMethod*	*ManufacturingOperation*
hasPlatform	*ManufacturingOperation*	*Platform*
hasProcess	*ManufacturingOperation*	*Process*
hasProductionMethod	*ProcessPlan*	*ProductionMethod*
hasSetup	*ManufacturingOperation*	*Setup*
hasResourceTool	*Setup*	*ResourceTool*
hasResourceToolOrientation	*Setup*	*ResoureToolOrientation*
hasWorkpieceOrientation	*Setup*	*WorkpieceOrientation*
hasStep	*ManufacturingProcess*	*Step*
usesResourceTool	*Step*	*ResourceTool*
involvesOperatorIntervention	*ManufacturingOperation*	*OperatorIntervention*
involvesFeature	*ManufacturingOperation*	*Feature*
isFollowedBy	*Operation*	*Operation*
isPrecededBy	*Feature*	*Feature*
hasNumberOfOperatorIntervention	*ManufacturingOperation*	*double*
requiresPrecedingFeatureStatus	*ManufacturingOperation*	*double*

It must be noted that all relations are assigned to the super classes with appropriate cardinality from which all subclasses inherit the relations. The domain of the relations signifies that every subject of the statements must belong to the class extension of the indicated class description (W3C., 2004). Furthermore, the range of the relations signifies that their values must belong to the class extension of the class description or to data values in the specified data range (W3C., 2004).

The crucial aspects of modelling the ontology are:

1. Key concepts are identified and defined with semantics that are generic for any types of manufacturing operations and their sequences. For example, in Figure 41 the concepts *Manufacturing Operation, Platform,* and *Setup* are some of the key concepts identified to capture manufacturing knowledge and build the ontology. To carry out a turning operation described in Table 11, it will require a *Platform* which is the combination of *Machine* and *Fixture* along with a *Setup* of cutting tools and the part orientation. Furthermore, the concept feature is identified for correct sequencing of operations. For example, in the process plan, the vane welding operation cannot be performed without the vane attachment feature being created;
2. Based on the key concepts, relationships that would enable to capture the knowledge of various operations with their probable sequences are specified as shown in Table 13. For example, the relationship *hasPlatform* is defined between *Manufacturing Operation* and *Platform*;
3. The semantics of the concepts are explicitly defined, making the model devoid of inconsistencies and incoherencies For example; *Autonomous Operation, Semi-Autonomous Operation* and *Manual Operation* are defined as different types of *Manufacturing Operation*. This ensures the consistent capture and categorisation of different types of manufacturing operations with subtle semantic differences;
4. The semantic integrity of the model is further preserved through formal axioms. This assists in rigorously defining the concepts. It is an enabler to semantically capture the complex scenarios of operation sequences;
5. Inference rules are defined on top of the semantic axioms to help extract and infer knowledge from the knowledgebase. They also assist in defining semantics.

Manufacturing operation knowledge is the prime focus of this aspect of the research. Therefore, the primary concept, i.e., *Manufacturing Operation*, is required to be defined accurately. One of the most appropriate definitions for *Operation* from manufacturing perspective is: *"An event in Process Plan that has a unique Setup"* (Usman., 2012). According to the definition, any change in the *Setup* results in the change of the *Operation*. This includes any manual intervention to the machine, such as changing the orientation of the tool. However, this would not hold truth for the following scenarios:

1. A *Manual Operation* requires mandatory multiple human interventions (fitting and bonding, welding operations as explained in the process plan). Therefore, a change in *Setup* including any manual intervention to the machine, such as changing the orientation of the tool or *Setup* would be classed as a different operation;
2. In case of *Autonomous Operation*, changes to *Setup* are automated. Here, once the *Machine Tool* and the *Fixture* has been setup, there is no need for any manual intervention unless there is an emergency (the milling operation in Table 11);
3. In scenarios when there are two different operations without any change in the *Setup*, e.g., in welding where the components are tack welded, followed by the final weld as described in the process plan in Table 11.

Furthermore, some of these manual interventions can be automated depending on the process and therefore categorised differentially. From the definitions, it can be understood that *Operator Intervention* is mandatory in case of *Semi-Autonomous and Manual Operations*. For *Autonomous Operations*, *Operator Intervention* may only be required for loading or unloading of a part. This scenario is captured via the *involvesOperatorIntervention* relation with the *Operator Intervention* concept. The cardinality of 1 to 0...* ensures the selective exploitation of the relation based on the type of *Operation*. A key aspect that differentiates between *Manual Operation* and *Semi-Autonomous Operation* is the number of times that the operator intervention is required. This is captured through the *hasNumberOfOperatorIntervention* relation. Thus, based on the above mentioned criteria, every *Manufacturing Operation* can be classified as one of the three types.

Within the model, the concepts of *Platform* and *Setup* are the most critical. They provide the route for capturing the operation sequencing knowledge. This is further detailed in the next Section. It is understood that each operation may involve different *Manufacturing Process*, such as Machining, Assembly and Inspection, etc. (shown in Table 11). Each process may involve different steps requiring different *Resource Tools*. For example, a machining process can be a milling, turning or drilling process which uses different *Resource Tools* and hence requires different Steps. Thus, every step is a sub-process for a specific process and a change in Step is signified by the change in the *Resource Tool*. This scenario is captured in the model through the *hasStep* and *usesResourceTool* relations (shown in Figure 41).

The creation, removal or validation of certain features is captured through involving the relation between *Manufacturing Operation* and *Feature*. This is a crucial relationship dictating the correct operation sequencing. Further, the state of the feature at every operation is considered as a requirement to adjudge whether an operation could be performed non-sequentially. This is explained in the next sub-section in detail.

5.2.2 Sequencing

The key concepts, their relationships as well as the categorisation of different operations, are defined in the previous sub-sections. However, one of the primary objectives of the proposed model is to capture and share the operation sequencing knowledge. The different sequences that can occur within a *Process Plan* defining the *Production Method* are described in Figure 40. The semantically rigorous model should be able to capture these sequences. Therefore, a proper capture of the underlying semantics of the concepts and constraints that constitute towards the change from one operation to another is required. A consideration in Figure 40 revealed that few scenarios are mutually complimentary with each other, considering the conditions dictating them are similar. Hence, Table 14 shows the final set of sequencing scenarios.

Table 14Final list of sequencing

Sequence	Precedence	
1	Autonomous Operation	Autonomous Operation
2	Manual Operation	Manual Operation
3	Semi-Autonomous Operation	Semi-Autonomous Operation
4	Autonomous Operation	Manual Operation

5	Autonomous Operation	Semi-Autonomous Operation
6	Manual Operation	Semi-Autonomous Operation

To address the different sequencing scenarios, two critical concepts of *Platform* and *Setup* are key attributes. They are the main drivers that dictate the change of operations. Figure 42 and Figure 43 reveal the scenarios which are dependent on *Platform* and *Setup* respectively. It must be noted that the scenarios which are dependent on the *Platform* do not require any mandatory human intervention in contrast to those dependent on *Setup*. The types of operations have the same annotation as in Figure 40.

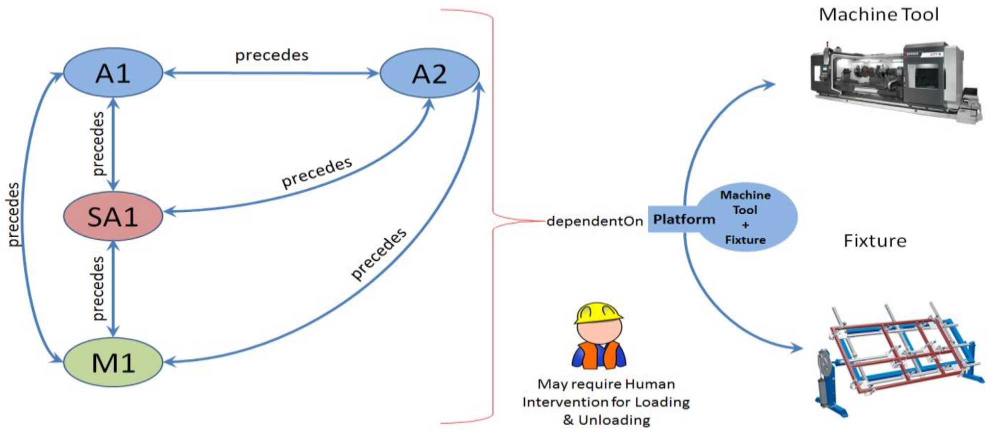


Figure 41 1st attribute contributing towards changes in Operation.

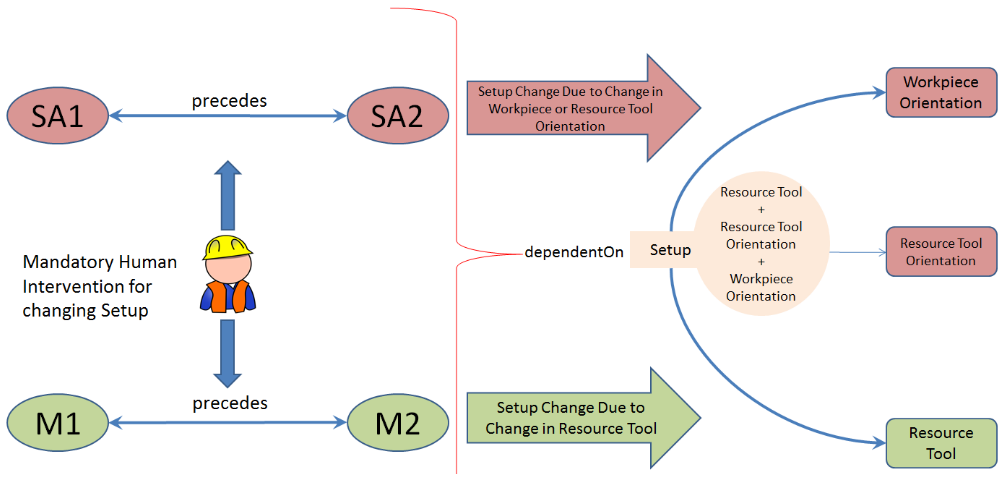


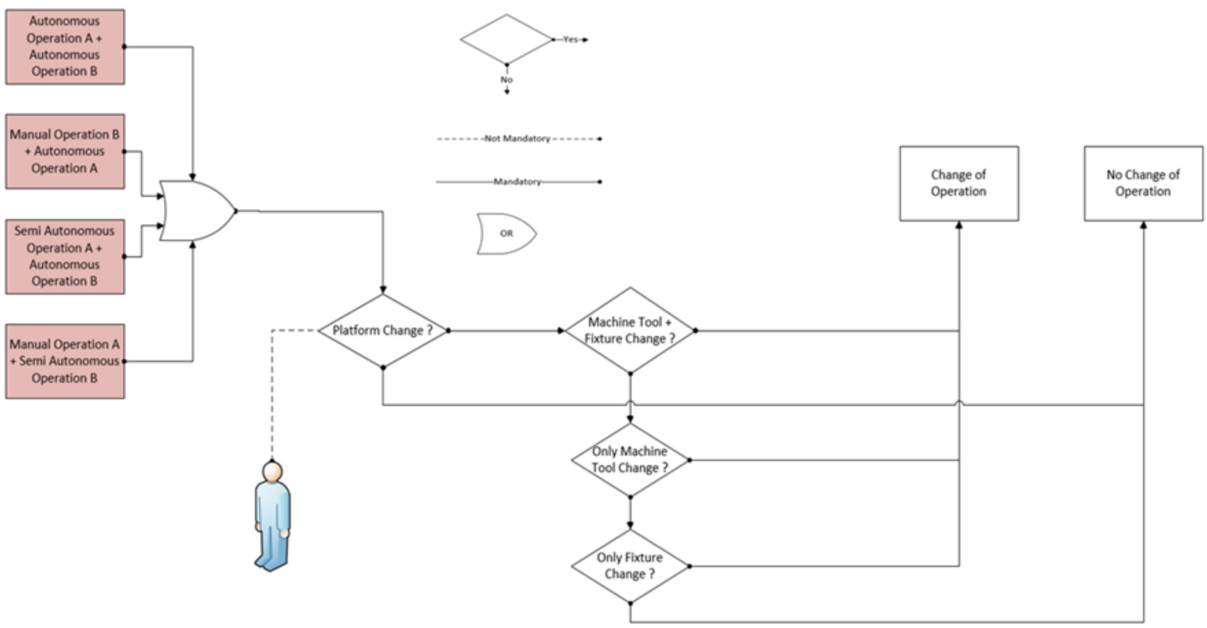
Figure 42 2nd attribute contributing towards changes in Operation

The detailed explanation of the operation sequences can be understood by following the flowchart in Table 15 and Figure 44. Table 15 describes all the possible combinations of two different Manufacturing

Operations: Manufacturing Operation A and Manufacturing Operation B. The flowchart in Figure 38 progresses from the left to right and illustrates the different scenarios and the criteria that dictate the change from one operation to another. The shaded rectangular boxes represent a combination of two processes while the rest portray individual processes. Any decision step is represented through the diamond boxes. The formal procedure shown in this flowchart is the key source for modelling the ontology to capture the operation sequencing knowledge.

Table 15 Combination of different types of Manufacturing Operations

	Manufacturing Operation A	**Manufacturing Operation B**	**Combination**
Operation Type	Autonomous Operation	Autonomous Operation	Autonomous Operation A + Autonomous Operation B
	Autonomous Operation	Semi-Autonomous Operation	Autonomous Operation A + Semi-Autonomous Operation B
	Autonomous Operation	Manual Operation	Autonomous Operation A + Manual Operation B
	Semi-Autonomous Operation	Semi-Autonomous Operation	Semi-Autonomous Operation A + Semi-Autonomous Operation B
	Manual Operation	Manual Operation	Manual Operation A + Manual Operation B



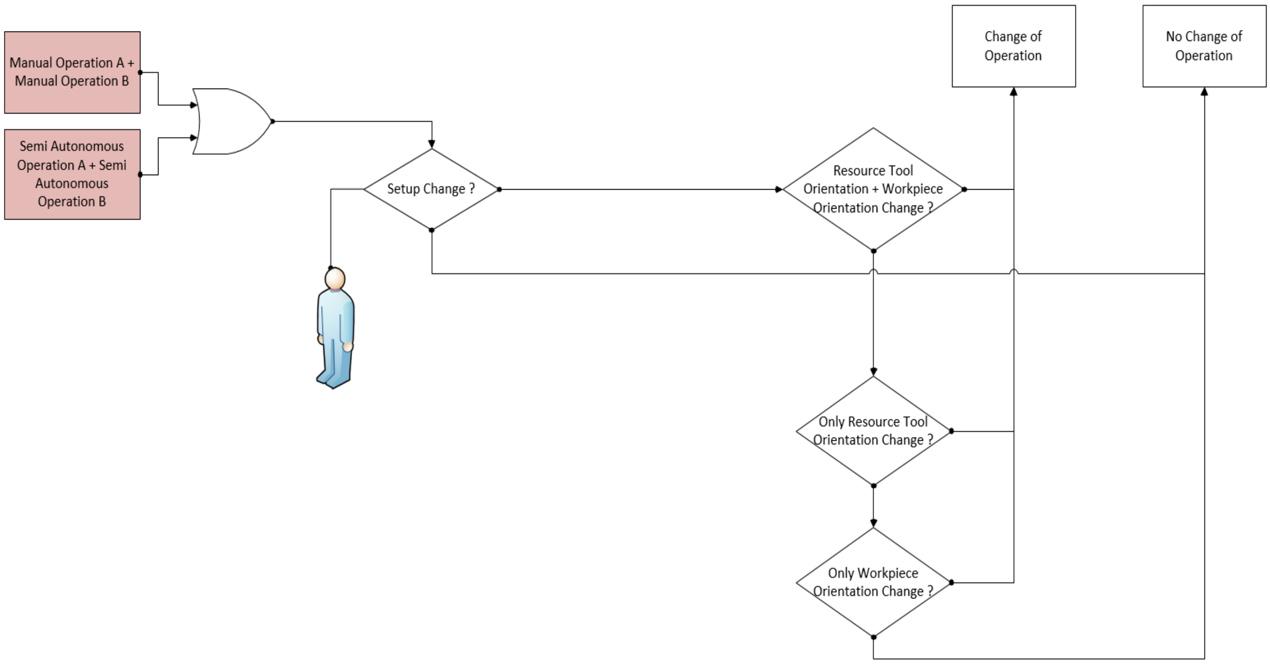


Figure 43 Flowchart for modelling process to capture sequencing knowledge

The above modelling can capture different operation sequences. However, it falls short in identifying the correct order of sequence. This is because the correct sequence further depends on the *Feature* that is involved at every operation. In a manufacturing environment, every operation always involves one or all of the following:

1. Creation of new features
2. Modification of existing features
3. Removal of existing features
4. Verification of features

This brings an extra layer of uniqueness and also dictates the correct sequence of operations. It is elaborated through the following example in Table 16 which shows a few different operations being carried out in order to manufacture an aero engine fan case assembly along with corresponding features that are involved these operations.

Table 16 Operations and features

Operations	Features
Inner Ring Machining Operation	Platform
Inner Ring Machining Operation	Stub
Stub Machining Operation	Knife Edge

An incorrect form of sequencing is illustrated below. In Figure 39, it can be seen that feature “Platform” is worked on in OP10. This is the first operation in the sequence as the other features “Stub”, “Knife Edge” are worked on after OP 10. However, in Figure 45, OP10 takes place at the end of the sequence which is not possible, considering the feature “Stub” cannot be created or worked on without the feature “Platform”. Therefore, the correct sequence is shown in Figure 46.

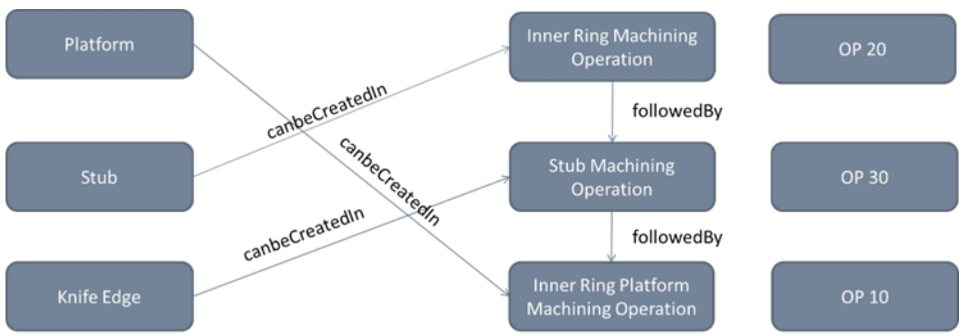


Figure 44 An incorrect sequence

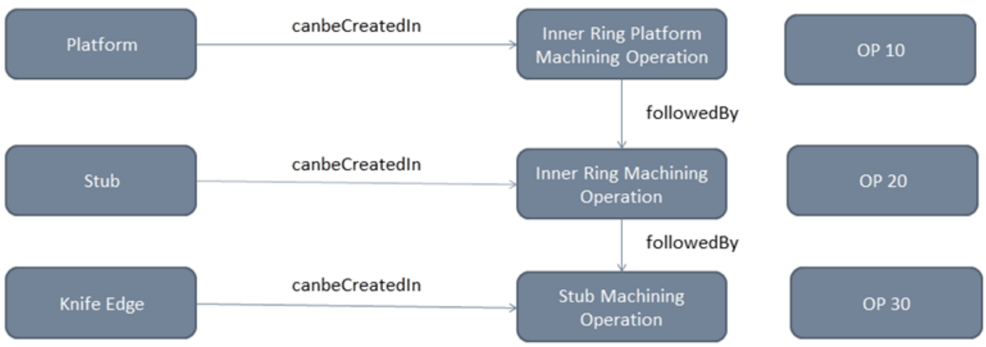


Figure 45 A correct sequence

In the manufacturing environment, an unique scenario arises where different operations can be performed out of sequence or in parallel to each other. This is particularly prevalent for assembly operations since some aspects of the operations can still be carried out even without the previous operation being complete. For example, the preparation of panels for a "bonding" operation can be carried out while a "fitting" operation is being performed. The following example shows a part of a process plan for an aero engine fan case assembly where some of the operations that can be performed in parallel or any order while the rest have an ordered sequence. In Figure 47:

1. OP 20 to OP50 can all be performed in any order or sequence but it is essential that all these operations are complete before OP60 can be performed.
2. Similarly OP70 has to be performed in sequence after OP60 and the group of parallel operations (OP20 – OP 50) has to be performed after OP10.

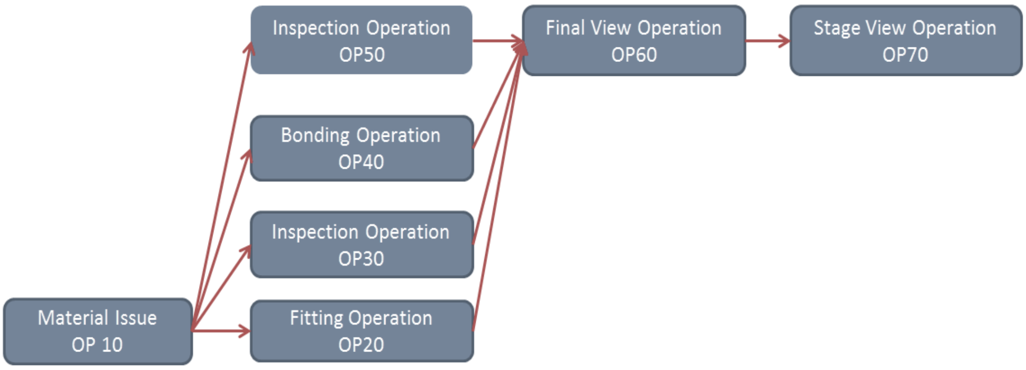


Figure 46Operations that can be worked in parallel

It is identified that the status of features being worked at that specific point of the operation allowed this to take place. Each of these operations did not require the preceding feature to be completed in order to be carried out. Figure 48 shows the modelling procedure for this scenario. Therefore, the proposed model is modified accordingly as follows:

1. A data property relation (*requiresPrecedingFeatureStatus*) is introduced to capture the status of the features at every *Manufacturing Operation*;
2. Numerical values are assigned to represent the state of the feature as every creation, modification and evaluation of any feature at particular operation adds some value to the entire product;

3. The maximum value is only added to a product when a feature is entirely completed and the state of the feature is attributed with a numerical value of 1. Any partial feature is attributed with values between 0 and 1. This depends upon the state of completion of the particular feature and the value accretion to the entire product.

Thus, depending on the value of *requiresPrecedingFeatureStatus* at every operation, the model infers if operation can be performed simultaneously. Any *Manufacturing Operation* that has the value for *requiresPrecedingFeatureStatus* to be less than 1 would enable the model to infer that the particular operation is open to be performed simultaneously to another.

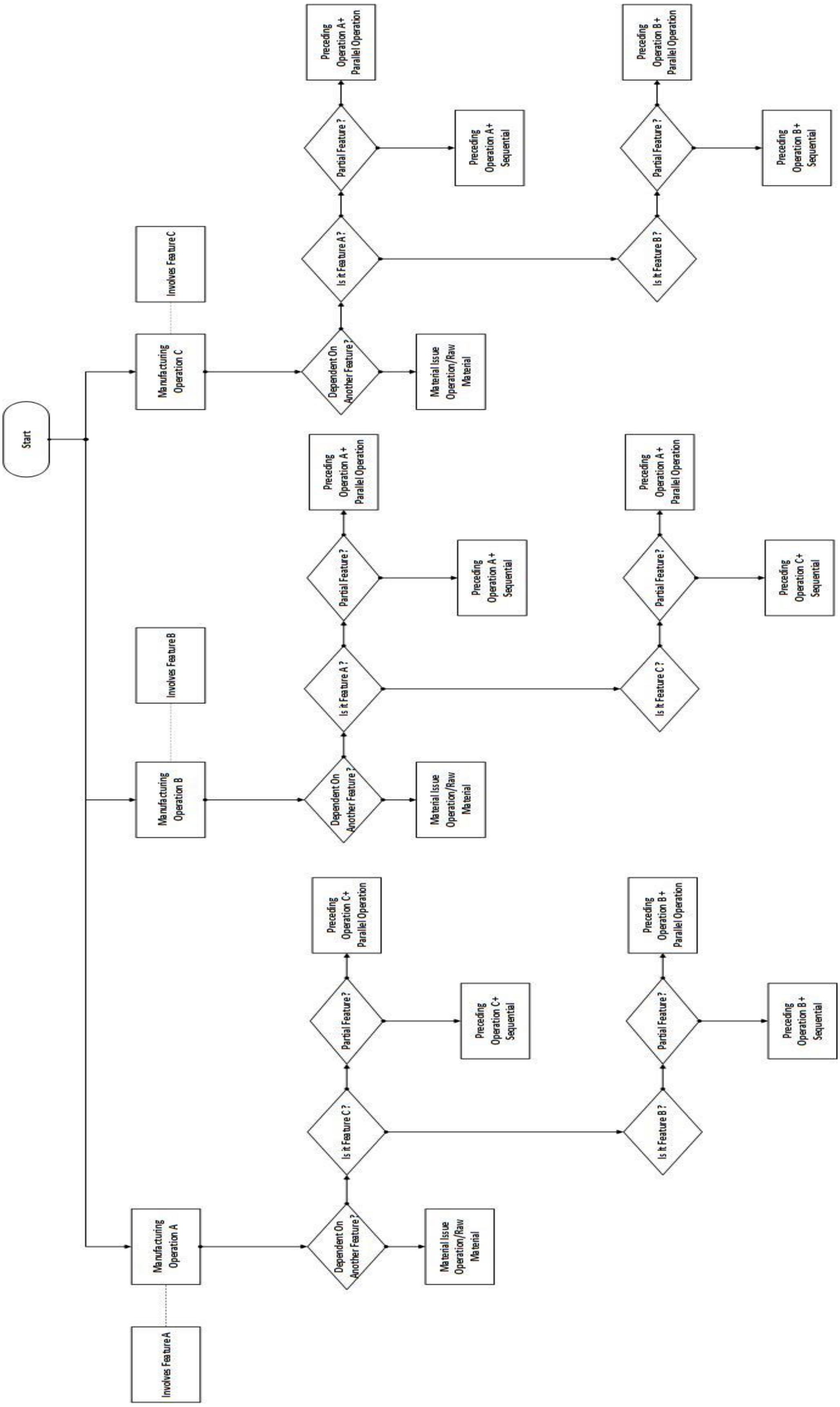


Figure 47Flowchart for modelling process of operation sequence with features and parallel operations

5.3 Implementation and Formalisation of the Model

This section elaborates the implementation procedure of the model. The overall high level implementation framework has been explained in Section 3.2.3.

The first step in the implementation of the framework is to identify the core concept that enables to capture the above mentioned knowledge. The requirement of this step is achieved through studying various process plans, manufacturing instruction documents, manufacturing drawings and through crucial input from the domain experts during the industrial study. Based on the identified core concepts and their relationship, the UML model is constructed. In the final step, the model is formalised into a heavyweight ontology using the description logic based language OWL DL. The formalisation procedure of the model by defining the classes, relations and restrictions have been carried out following the same procedure as explained in Section 3.3.3. OWL DL has been used as the formalisation language with the addition of SWRL for assigning complex rules. The final step of the implementation process is the experimental verification of the model.

5.4 Experimental Validation

In this section the proposed PLO's capability to capture the knowledge pertaining to manufacturing processes and operation sequences have been experimentally verified. This experimental verification is essentially based on a case study for the manufacture of aero engine fan case assembly. The process plan for the manufacture has been elaborated in Section 5.2. The experiment validates that the model is able to capture the requirements stated earlier. This includes the following

1. Verification of semantic capture through categorisation of different types of operations.
2. Inference of new knowledge through identifying the correct sequence of operations
3. Identification of the operations that can be carried out in parallel.

The structure of the knowledge base has been created by defining the different classes, relations, constraints, rules and axioms as described in Section 5.2 and Section 5.3. The knowledge base is then populated with several instances of operations and other attributes required in the manufacturing of an aero engine fan case assembly. The asserted instances and their corresponding classes are shown in Table 17.

Table 17 Asserted instances and their corresponding classes

Class	Instances
Fixture	Milling Fixture1 Milling Fixture2 Turning Fixture1 Turning Fixture2 Welding Fixture1 Welding Fixture2 CMM Fixture1 CMM Fixture2 Fitting Bonding Fixture1
Machine Tool	Machining Centre1 Machining Centre2 Machining Centre3 Machining Centre4 Welding Bode1 Welding Bode2 Welding Bode3 Welding Bode4 CMM1 CMM2 Fitting Bonding Bay1
Resource Tool	Milling Cutter1 Milling Cutter2 Milling Cutter3 Turning Cutter1 Turning Cutter2 Tack Welding Torch1 Tack Welding Torch2 Final Welding Torch1 Final Welding Torch2 CMM Probe1 CMM Probe2 Torque Controlled Power Tools Blanking Tools Inspection Tools OGV Assembly Tools
Resource Tool Orientation	Facing Inwards At An Angle Facing Outwards At An Angle Horizontal Vertical Variable
Workpiece Orientation	Resting On Bottom Face Resting On Top Face

	Vertical At BDC Vertical At TDC
Feature	Platform Stub Knife Edge Abutment Face Position Joint Locater Joint Material Issue Liners Panels

Figure 49 shows the assertion of some these instances and their corresponding properties into the knowledge base.

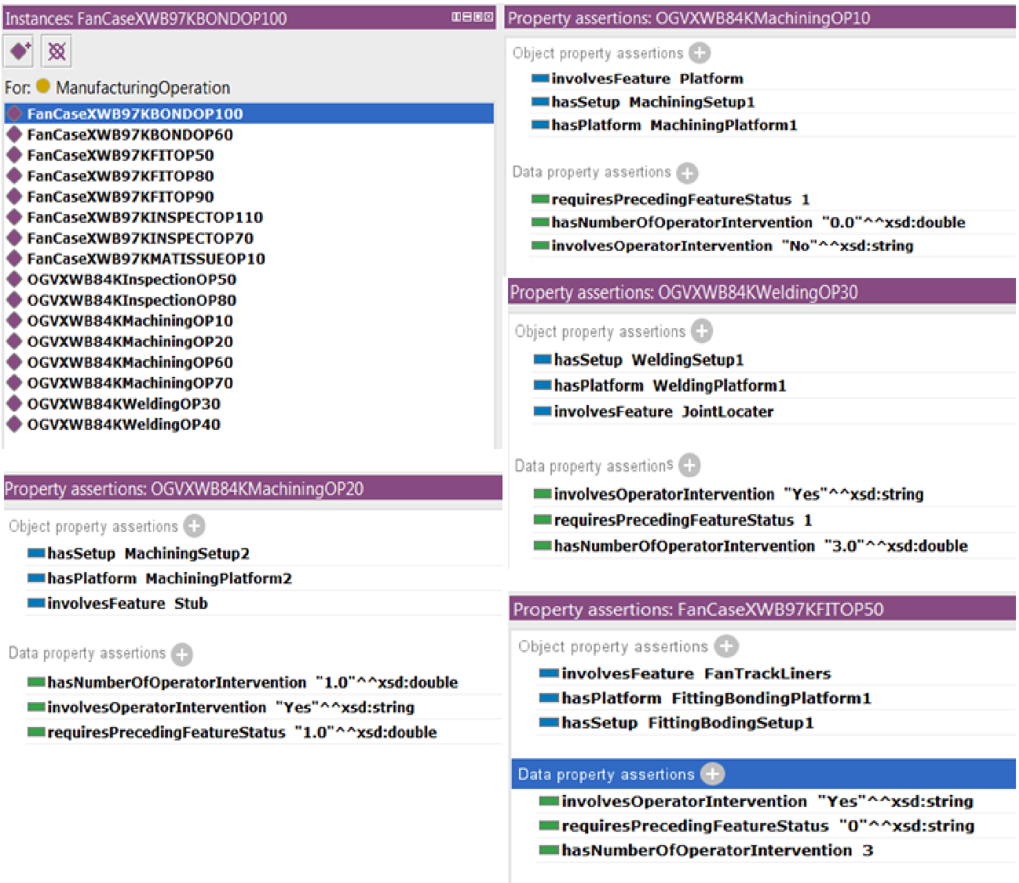


Figure 48Assertion of classes/concepts in Protégé

An important aspect of OWL which is required to be taken into account during the assertion is that OWL works on Open World Assumption (OWA). This implies that the system will not be able to infer something unless it is explicitly specified. For example, if something is specified to be not true does not mean that it would be false rather it is simply inferred as unknown. Therefore the concerned instances have been assigned as different individuals.

1. With the knowledge base being populated with all the instances, it is now reasoned using DROOL rule engine and then queried using the following query.

SQWRL Query:

```
AutonomousOperation(?AutonomousOperations) -> sqwrl:select(?AutonomousOperations)

SemiAutonomousOperation(?SemiAutonomousOperations) -> sqwrl:select(?SemiAutonomousOperations)

ManualOperation(?ManualOperations) -> sqwrl:select(?ManualOperations)
```

Figure 49 Query to classify different operations

Figure 51, shows the result of querying the knowledge base for the list of *AutonomousOperations*, *ManualOperations* and *SemiAutonomousOperations*. Therefore, the test validates that the model is able to categorise all the types of operations.

AutonomousOperations
:OGVXWB84KInspectionOP50
:OGVXWB84KMachiningOP10

SemiAutonomousOperations
:OGVXWB84KMachiningOP20
:OGVXWB84KMachiningOP70
:OGVXWB84KMachiningOP60

ManualOperations
:OGVXWB84KWeldingOP40
:OGVXWB84KWeldingOP30
:FanCaseXWB97KBONDOP60
:FanCaseXWB97KFITOP90
:FanCaseXWB97KFITOP80
:FanCaseXWB97KFITOP50
:FanCaseXWB97KINSPECTOP110
:FanCaseXWB97KINSPECTOP70
:FanCaseXWB97KBONDOP100

Figure 50Results of querying to classify different operations

2. The second set of test was carried out to infer the correct sequences of the operations required to produce a conforming product. The following query is used to retrieve the correct sequence of operations based on populated instances.

SQWRL Query:

hasSetup(?followingOP, ?followingSetup) ^ isFollowedBy(?precedingOP, ?followingOP) ^ hasSetup(?precedingOP, ?precedingSetup) ^ hasPlatform(?precedingOP, ?precedingPlatform) ^ hasPlatform(?followingOP, ?followingPlatform) -> sqwrl:select(?precedingOP, ?precedingPlatform, ?precedingSetup, ?followingOP, ?followingPlatform, ?followingSetup)

Figure 51Query to retract the correct sequence of operations

Figure 53, shows the results for query regarding the sequences or precedence of operations. The result is based on the semantic criterions which have been discussed before and verifies that the model is able to capture those semantics. This figure shows the correct sequences of operations that can take place and also infers the corresponding *Platform* and *Setup* at each operation. Thus, the results verify the model's capability to infer the correct sequence of operations.

precedingOP	precedingPlatform	precedingSetup
:FanCaseXWB97KFITOP50	:FittingBondingPlatform1	:FittingBondingSetup1
:FanCaseXWB97KBONDOP60	:FittingBondingPlatform1	:FittingBondingSetup2
:FanCaseXWB97KFITOP80	:FittingBondingPlatform1	:FittingBondingSetup4
:OGVXWB84KMachiningOP20	:MachiningPlatform2	:MachiningSetup2
:OGVXWB84KMachiningOP10	:MachiningPlatform1	:MachiningSetup1
:FanCaseXWB97KINSPECTOP70	:FittingBondingPlatform1	:FittingBondingSetup3
:OGVXWB84KMachiningOP70	:MachiningPlatform6	:MachiningSetup4
:OGVXWB84KMachiningOP60	:MachiningPlatform5	:MachiningSetup3
:FanCaseXWB97KFITOP90	:FittingBondingPlatform1	:FittingBondingSetup5
:OGVXWB84KWeldingOP40	:WeldingPlatform1	:WeldingSetup2
:OGVXWB84KWeldingOP30	:WeldingPlatform1	:WeldingSetup1

followingOP	followingPlatform	followingSetup
:FanCaseXWB97KBONDOP60	:FittingBondingPlatform1	:FittingBondingSetup2
:FanCaseXWB97KINSPECTOP70	:FittingBondingPlatform1	:FittingBondingSetup3
:FanCaseXWB97KFITOP90	:FittingBondingPlatform1	:FittingBondingSetup5
:OGVXWB84KMachiningOP60	:MachiningPlatform5	:MachiningSetup3
:OGVXWB84KMachiningOP20	:MachiningPlatform2	:MachiningSetup2
:FanCaseXWB97KFITOP80	:FittingBondingPlatform1	:FittingBondingSetup4
:OGVXWB84KWeldingOP30	:WeldingPlatform1	:WeldingSetup1
:OGVXWB84KMachiningOP70	:MachiningPlatform6	:MachiningSetup4
:FanCaseXWB97KBONDOP100	:FittingBondingPlatform1	:FittingBondingSetup6
:OGVXWB84KInspectionOP50	:InspectionPlatform1	:InspectionSetup1
:OGVXWB84KWeldingOP40	:WeldingPlatform1	:WeldingSetup2

Figure 52Inference of correct operation sequence

3. As it has been described in the previous sections that the model is able to identify and infer the operations that can be performed in parallel with each other based on the populated knowledge. The following query is used to retrieve the operations that can be carried out in parallel.

SQWRL Query:

isParallelWith(?OP, ?isParallelToOP) -> sqwrl:select(?OP, ?isParallelToOP)

Figure 53Query to identify parallel operations

Figure 55 shows the operations that can be carried out in parallel to each other. Therefore, the test verifies the model's capability to identify the operations allowed to be carried out in conjunction with other.

OP
:FanCaseXWB97KINSPECTOP70
:FanCaseXWB97KFITOP80
:FanCaseXWB97KFITOP90
:FanCaseXWB97KFITOP50

isParallelToOP
:FanCaseXWB97KFITOP80
:FanCaseXWB97KFITOP90
:FanCaseXWB97KBONDOP100
:FanCaseXWB97KBONDOP60

Figure 54Operations allowed to be carried out of sequence

5.5 Summary

This chapter has elaborated the utilisation of PLO to capture the manufacturing operations and its sequencing knowledge. The various complexities involved in the sequencing of manufacturing operations in a production environment are discussed first. It followed by elaboration of the requirements for a robust model that would provide the knowledge of various operations to the production engineers. A semantically enriched core manufacturing ontology as an extension of PLO has been introduced. This enables formal capture and sharing of operation sequencing knowledge. A set of core manufacturing concepts and relations have been identified and formally defined to model the manufacturing operations and sequencing knowledge. The ontological formalisation of the proposed model is carried out using Web Ontology Language (OWL). A secondary layer of rules and axioms using SWRL has been defined to address complex scenarios and infer new knowledge. And lastly, PLO was validated for its capture of manufacturing operation and its sequencing knowledge.

6. Ontology Evaluation

This chapter explains the evaluation of the proposed ontology. In the previous chapter PLO was verified through experimentation and a case study. It had portrayed that the proposed PLO is able to fulfil the requirements of its designed applications. Furthermore, it was able to handle challenges of a real world scenario. However, the PLO is required to be measured against a wider framework of similar type of work. The benefits of the proposed PLO over the existing prevalent models are required to be highlighted. Thus, the evaluation of the PLO encompasses all these aspect. In this chapter, the various ontology evaluation techniques are discussed in Section 6.1. This is followed by the evaluation of PLO in Section 6.2 and summarisation in Section 6.3.

6.1 Ontology Evaluation Methods

Within the world of ontological engineering, there is a general disparity on the best possible way to evaluate ontologies. Various researchers have proposed different methods of evaluation. (Staab, 2009) elaborated on the concepts of verification and validation. According to them, verification of ontology entails ascertaining its quality. This comprises of checking whether the ontology is devoid of any inconsistencies and has all the required concepts. Validation on the other hand determines if the correct ontology has been built. This is established through the capability of the ontology in meeting the developed application requirements. Both of these have already been achieved in the previous experimental section. However, (Ahmad, 2017) argued the need of a broader validation requirement that determines the improvement of the proposed model over the prevalent models.

(Obrst, 2007), first pointed out the need to define a systematic approach to evaluate ontologies rather than having an approach that just satisfies the requirements sufficiently. This was to ensure that the development of the ICT systems is more methodical. Their view for ontological evaluation procedures was based on those from biomedical field. It comprised of natural language evaluation, application evaluation and comparison of the domain data with the developed ontology. However, the adoption and reuse of the ontology was their recommendation as the best method for evaluation. (Brank, 2005), (Hlomani, 2014) defined ontology evaluation as a measure of ontology quality on certain set criteria's based on proposed applications. They carried an extensive survey on the existing methodologies of ontology evaluation. Based on their survey the following Table 18 shows the different evaluation methodologies.

Table 18Ontology Evaluation Methodologies

Evaluation Method	**Short Description**
Application Driven	This methodology involves evaluating the effectiveness of the ontology based on an use-case or an application. It is very difficult to generalise the results from this methodology. This is because the results from one application may not necessarily hold true for another application. Moreover, comparing a large number of ontologies is challenging and laborious process.
User Based	This evaluation method is based on the experience of the user that utilises this ontology. Essentially the metadata from the viewpoint of the ontology creators are compared against those from the end users. Thus, the subjective information of the ontology is evaluated through this method. The challenging aspect of this method is identifying the right set of users.
Data Driven	As the name suggests, this method entails comparing the existing data from the domain it is trying to model with the ontology. The most common methodology to carry this out is by comparing the concepts from the domain with that of the ontology. However, the drawback of this method is that these concepts evolve through addition of new ones due to the dynamism of the domain knowledge.
Gold Standard	In this methodology, the ontology is compared against another ontology which is considered as a “gold standard”. A “gold standard” is essentially an ontology which is structurally sound, expressive and comprehensive within the concerned domain. But question arises on the evaluation of the “gold standard” ontology itself.

While most of the methodologies suffer from being prone to subjectivity, the data driven approach lacks the appreciation for the dynamic nature of the domain knowledge. The matter of subjectivity is an inherent part of development, as ontology is a conceptualisation that only endeavours to approximate the real world. Thus, an ontology developer invariantly dictates the developed ontology with their own preferences and expertise. The notion of subjectivity cascades onto the evaluation methodologies as well, since the assessment of the evaluators are based on their own conceptualisation. Therefore, the evaluation methodology should thrive to measure how far the approximated conceptualisations are from the real world (Ahmad, 2017). (Gómez-Pérez., 2001) (Vrandečić., 2009) (Hlomani, 2014) (Bandeira, 2016) proposed several criteria metrics for the above mentioned methods as shown in Table 19.

Table 19 Criteria's for Ontology Evaluation

Criteria	Short Description
Accuracy	Measure of the extent of agreement between the expert knowledge and those asserted into the ontology.
Adaptability	It measures the flexibility of the ontology. Essentially it is the level of easiness with which the ontology can be extended and used for different applications.
Clarity	It measures the efficiency of the ontology to easily communicate the meanings of the concepts.
Cohesion	The level of modularity which dictates the relation between the classes.
Competency/Completeness	This decides the extent of the ontology in covering the regime of the concerned domain and the concepts they comprise.
Computational Efficiency	Measure of the speed by which the ontology reasoners are able to infer knowledge.
Conciseness	This ascertains the number of redundant concepts with regards to the domain modelled. The smaller the amount of such concepts ensures

	minimalist ontological commitment.
Consistency/Coherence	This ensures less contradictions and better consistency within the ontology.
Organisational Fitness	The deployability of the ontology in an organisational application environment is dictated through this criterion.

6.2 Evaluation of PLO

The “Gold Standard” methodology has been used for the evaluation of PLO based on the following criteria’s

- Accuracy
- Adaptability
- Consistency
- Conciseness
- Completeness

The DFM, MCCO and ARO ontologies have been used as the “Gold Standard” ontologies for comparison with PLO. Extensive descriptions of the ontologies are available which makes them ideal for comparison. From the reviewed literature in Section 2, it can be seen that the above mentioned ontologies are the most relevant ontologies for product lifecycle. Most of the other ontologies were either developed for more generic purpose or constrained to a specific domain with very few at a core or reference level. Further, these ontologies have been highly cited by other authors. The above criteria have been chosen as sufficient information was available to carry out the evaluation. The following Table 20 lists the evaluation of PLO in comparison with the selected “Gold Standard” ontologies, based on the aforementioned criteria.

Table 20 Evaluation of PLO against different ontologies and criteria's

Criteria	Evaluation
Accuracy	**DFM, MCCO and ARO**: Case studies have been used to evaluate the accuracy of these ontologies. **PLO**: The successful modelling of the case study based on real world scenario portrayed the preciseness of the ontology.
Adaptability	**DFM, MCCO and ARO:** There has not been any indication of any extensibility of the DFM ontology. MCCO and ARO on the other hand have shown some level of extensibility through inclusion of new classes. However, their adaptability has been limited to their specific domain with no scope of being adapted for beyond. **PLO**: The experimental verification and the case study have portrayed the adaptability of PLO through inclusion of new classes from different and multiple domains. These new classes were added to model application specific ontologies from different domains of the product lifecycle.
Clarity	**DFM, MCCO and ARO:** There are only limited explanations of the concepts which are primarily understood by domain experts. Furthermore, there has been very limited segregation of the concepts which increases the obscurity. This transcends across to different relations between the concepts. Apart from few concepts and relations the majority within MCCO and ARO have been clearly defined. **PLO**: It has been ensured that all the concepts and relations are defined unambiguously. The names of the concepts have been based on international and industry standards which are agreed by the wider community.

Competency/Completeness	**DFM, MCCO and ARO:** MCCO comprises of a comprehensive number of concepts that can model the machining domain with some process planning concepts. DFM and ARO are not extensively complete in context of the domain they model. For example, ARO is devoid of concepts that can model welding knowledge. Similarly, DFM ontology was developed to encompass all manufacturing process but had only concentrated on the welding domain concepts. **PLO:** In terms of completeness, PLO is more comprehensive as some of the concepts have been derived from the "Gold Standard" ontologies. The newly defined concepts were proposed to fill the gaps from the existing ontologies. The PLO is an enhancement and thus bears concepts that can model multiple domains in its entirety.
Conciseness	**DFM, MCCO and ARO:** Apart from MCCO, not all concepts within the ARO and DFM have been used for experimental verification. However, this does not essentially construe that they were redundant concepts but rather those concepts make the ontology more adaptable. **PLO**: Similar to DFM and ARO, not all concepts have been used for verification. Also, the minimal number of constraints required to model the concepts ensures easier commitment approaches.
Consistency/Coherence	**DFM, MCCO and ARO:** MCCO and ARO concepts are consistent in their approach as they were developed from the perspective of knowledge sharing. Furthermore, their experimental validation has shown this. DFM on the other hand has few classifications where the consistency is questionable. **PLO**: The consistency of the PLO has been experimentally verified, especially with regards to the welding concepts.

It must be noted that the ontologies that have been chosen for evaluating PLO against, are the most closest that can be found in the literature with similar objectives. A characteristics of PLO that is worth highlighting is the declarative and expressiveness of the defined relations. This makes it easier for the ontologists to understand the model and further define queries. The PLO concepts are comprehensive and have the capability to model multiple product lifecycle domains. Thereby, placing it as true core ontology from which very specific domain ontologies can easily be built. Although all the domains have not been modelled extensively as they are beyond the scope of this work, but PLO provides the relevant concepts from which any domain ontology can be created. The welding hierarchy model within PLO portrayed the consistency of the model and its benefits over the standards. Similarly, the operation sequencing model within PLO is unique in its own accord. The closet models that had tried to model operation sequences were the PSL model and that proposed within MCCO. However, their model has shortcomings in terms of the granularity with which PLO is able to achieve. It was impossible to find one single ontology in the literature that had the capability to model all the three following aspects.

1. To share knowledge across multiple product lifecycle domains
2. To establish a semantically enrich welding hierarchical model and an ontology.
3. To model the manufacturing operation sequencing knowledge.

Therefore, this in itself is a unique and novel trait of the PLO in its evaluation.

6.3 Summary

This chapter elaborated the evaluation of the proposed PLO to highlight its novelty. The research was experimentally verified and validated in an industrial environment. However, it is still required to be evaluated in terms of novelty and contribution to knowledge with respect to prevalent research works. Therefore, the proposed ontology has been evaluated using a specific method and different metrics' against the objectives set out. From the different methods found in the literature, the "Gold Standard" methodology have been utilised for evaluation. Based on the metrics, it was concurred that the PLO was more accurate, adaptable, consistent, concise and complete when compared to the most established ontologies found in literature. Furthermore, the evaluation led to conclude that there is an inexistence of a single model that achieves all the objectives laid out in this thesis. This makes PLO a novel model in its entirety.

7. Research Outcomes, Conclusions and Future Work

7.1 Research Outcomes and Conclusions

This research had focused on the development of a solution model that can be applied across the domains of design, machining, welding and inspection. The primary concentration has been towards enhancing the ability to share the knowledge residing within the different manufacturing domains to design engineers. The availability of the manufacturing knowledge at the disposal of the designer engineers enhances the design right first time. More specifically, addressing the knowledge sharing issues pertaining to the domains of machining, assembly, welding and inspection together with their process planning was one of the key objectives of this research.

The first step towards achieving the defined objectives was to identify the current state of the art in knowledge modelling and manufacturing ontologies. Therefore, a comprehensive literature review was carried out within the field of knowledge sharing across product lifecycle domains in Chapter 2. The extensive review assisted in ascertaining the various interoperability issues of ICT based systems for multiple product lifecycle domains. Furthermore, existing ontological models and methods for knowledge sharing have been explored in exhaustiveness. The research gaps were further strengthened from the industrial case study and discussed in Chapter 2. The identified research gaps could be overcome by achieving the following requirements as discussed in Chapter 3:

a) The different perspectives of the concepts were required to be captured. This implied that the semantics of the concepts along with their context and relations are required to be captured. This is to ensure interoperability across multiple product lifecycle domains.
b) A set of reusable and overlapping core concepts were required to be identified from the domains of design, machining, assembly, welding and inspection.
c) An appropriate formalisation language was required to capture the semantics of concepts from multiple product lifecycle domains. Additionally, they would be required to identify the similarities and differences between concepts from all these domains.

Therefore, based on the above mentioned requirements, a core set of concepts represented through a Product Lifecycle Ontology (PLO) was introduced in Chapter 3. The methodology followed to develop this model including the core set of concepts and their collective relationships has been extensively discussed in this chapter 3. PLO is core ontology and therefore its concepts are generic to encompass multiple product lifecycle domains. However, the requirements of the PLO resulted in defining new layers to capture the varying depths of meanings. Thus, several new specialisations of the product lifecycle concepts both at the core and the domain level were defined to capture the knowledge with higher granularity. PLO, a core ontology was constructed to be an intermediate layer and not a pure design or manufacturing domain ontology. The translational links and relations between the foundation concepts, PLO core concepts and the domain concepts were established. Furthermore, the PLO was utilised as a sematic base to develop application specific ontologies. *Feature* was identified as the primary concept for knowledge sharing across the domains of design, machining, assembly with welding and inspection. Furthermore, it has been exploited to develop the application specific design and manufacturing ontologies as shown in Chapter 3. The utilisation of *Feature* and their corresponding concepts elaborated the knowledge sharing aspect from a high level perspective. That is the share-ability of the combined knowledge from multiple domains to design. Thereby, it acted as a route to share knowledge across multiple product lifecycle domains.

In chapter 4, PLO has been portrayed to act as a semantically enriched welding ontology to achieve interoperability. The semantic inconsistency issues in welding standards were systematically investigated first. This was followed by the utilisation and extension of PLO to capture the welding knowledge. It showcased a new approach of capturing semantics of welding and joining concepts. This new methodology was used to resolve the semantic inconsistencies within and across welding standards. Furthermore, it was exploited to facilitate the knowledge sharing across welding domains that use different standards through their consolidation. The multifaceted capability of PLO is further showcased in Chapter 5. This was done via its exploration to capture manufacturing operations and their sequencing knowledge. The various complexities involved in modelling manufacturing knowledge was discussed using manufacturing process of an aero engine fan case assembly. PLO was then explored to model the different types of manufacturing operations and their sequences.

The informal description of the core concepts have been formalised using Web Ontology Language Description Logic (OWL DL). The formalisation process has been explained through the use of different namespaces, classes, properties, restrictions, rule and axioms. The Semantic Web Rule Language (SWRL) has been used as an added layer to model complex rules and axioms. Protégé ontology editor was then used for development as well as the implementation of the ontology as explained in Chapter 6. This was followed by experimentally verifying the ontology against various test scenarios. PLO was experimentally validated against the following categories

1. Semantic integrity i.e the consistency checking of PLO.
2. Ability to develop application specific ontologies
3. Identifying the route to share knowledge.
4. Retracting the manufacturing related knowledge.
5. Capture of welding related knowledge.
6. Consolidation of welding standards.
7. Capture the manufacturing process related knowledge.
8. Identification of correct sequences and concurrency of operations required to manufacture a product.

The proposed model was further validated in a real world production environment through an industrial case study. Through this case study, the models capability to retrieve the implications of design changes on the manufacturability of an aero engine OGV assembly was explored. PLO was used to retract the knowledge pertaining to the machining, assembly with welding and inspection for the OGV assembly. This retrieved knowledge was fed back to the designers to validate their design. And, finally in Chapter 7, PLO was evaluated for its novelty against the existing research. The "Gold Standard" methodology was used for evaluation using different metrics such as: accuracy, adaptability, consistency, conciseness and completeness. Based on the evaluation, the original hypothesis of,

"A formal core ontology can support knowledge sharing from machining, welding and inspection domains with product design by providing a common verifiable semantic base."

was proven. Consequently, the following novel aspects of PLO and this research framework can be concluded:

1. The multiple specialisations of the product lifecycle concepts both at the core and the domain level to capture the knowledge with higher granularity have been shown. The proper categorisation of concepts through the specialisation levels was portrayed to support the elimination of semantic mismatches.
2. PLO was shown to act as a semantic base that supports knowledge sharing from multiple manufacturing domains. This is one of the novel aspects of the research as a single model that has the capability to share knowledge across design, machining, welding and inspection domains is non-existent.
3. The core concepts from PLO were shown to be utilised and extended to capture the welding specific knowledge. A new novel methodology for welding and joining process categorisation was revealed. It was further explored to reconcile the semantic inconsistencies and incoherencies across multiple standards through their consolidation.
4. A new approach to categorize manufacturing operations was formulated using the core concepts of PLO. The postulated model's novelty lies in its capability to provide the correct sequences of operation and identify the concurrent operations.
5. The multidimensional capability of the model can potentially aid the manufacturing engineers for informed decision making during introduction of new products. Furthermore, the ability to portray the operation sequencing information is highly beneficial for the production planners to take prompt decision during production planning.

7.2 Recommendations for Future Work

The research work carried out revealed several directions of potential future works. Some of these are

1. To make the ontological model have an overarching applicability on broader domains, research could be carried out to utilise and extend PLO for other domains of the product lifecycle. Such as *Service*, *Repair*, *Maintenance, Disposal* etc.
2. Within the manufacturing domain, the concepts can be explored to encompass to other processes such as casting, non-conventional machining, additive manufacturing, moulding etc. A core set of concepts suitable for these domains can be explored to be defined from PLO.

3. The specialisation levels can be explored further to capture more varying depths of meanings. This might potentially be required when several other domains are considered.
4. The joining and welding processes model can be explored further for other processes. In this research, the model was used for only welding and its standards. Therefore, a research work for other joining processes and their standards can be carried out.
5. With respect to the manufacturing operation and its sequencing model, the consequences of making design changes on the operation sequences can be explored further. In this research, the model was used to categorise the different forms of operations and infer the correct sequence. However, the result of changing the design onto the types and sequences of operations can be further explored.
6. The core concepts for design can be explored further to explore other aspects of design engineering. These can potentially be areas such as aerodynamic analysis, thermal analysis, stress analysis etc.
7. The proposed approach could also be explored for its use in other business domains such as medical sciences and civil engineering etc.

References

Abdul-Ghafour, Samer, Parisa Ghodous, Behzad Shariat, Eliane Perna, Daniel Frey, Shuichi Fukuda, Georg Rock. *Ontology Development for the Integration of CAD Models in a Collaborative Environment Improving Complex Systems Today.* London: Springer, 2011.

Abramovici, Michael, Olaf C,. “Status and development trends of product lifecycle managementr systems.” *In IPPD Conference.* Wroclaw: Sieg, 2002.

Ahmad, Mussawar. *An ontology-based approach for integrating engineering workflows for industrial assembly automation systems.* PhD Thesis. Warwick: University of Warwick, 2017.

Ahmed Al-Ashaab, Karina Rodriguez, Arturo Molina, Mauro Cardenas, Joaquin Aca, Mohammed Saeed, Hassan Abdalla. “Internet-based collaborative design for an injection-moulding system.” *Concurrent Engineering:Research and Applications* 11, no. 4 (2003): 289–299.

Aifaoui, N., Deneux, D. and Soenen, R.,. “Feature-based interoperability between design and analysis processes.” *Journal of Intelligent Manufacturing* 17 (2006): 13-27.

Al-Ashaab, A H,. *A manufacturing model to capture injection moulding process capabilities to support design for manufacture.* Loughborough, UK: Loughborough University, 1994.

Ali, M M, Rai R, Otte J N, Smith B,. “A product lifecycle ontology for additive manufacturing.” *Computers in Industry* 105 (2019): 191-203.

AMICE. *CIMOSA – Open System Architecture for CIM.* 2. Berlin, Germany: Springer-Verlag, 1993.

Anderl, R., “Trends in product modelling.” *In: Proceedings of the 11th International Conference on Engineering Design.* Tampere University of Technology., 1997. 113-120.

Anjum, N. *Verification of Knowledge Shared across Design and Manufacture Using a Foundation Ontology.* Loughborough: Loughborough University, 2011.

Anouncia, S.M, Saravanan R,. “A knowledge model for gray scale image interpretation with emphasis on welding defect classification—An ontology based approach.” *Computers in Industry* 61 (2010): 742–749.

Antoniou, Grigoris. Frank Van Harmelen. *A Semantic Web Primer 2nd Edition.* Cambridge: MIT Press, 2008.

Antoniou, G, and Harmelen F V,. “Web Ontology Language: OWL.” In *Handbook on Ontologies*, by and R. Studder S. Staab, 91-110. Berlin:: Springer-Verlag, 2009.

Arthur, Jeffrey B. and Christopher L. Huntley. “Ramping up the Organizational Learning Curve: Assessing the Impact of Deliberate Learning on Organizational Performance under Gainsharing.” *The Academy of Management Journal* 48, no. 6 (2005): 1159-1170.

Babcock, Pamela. “"Lessons learned lead to new ideas about sharing information-in Shedding Light on Knowledge Management.” *In HR Magazine*, 2004.

Balogun, O., Hawisa, H. and Tannock, J.,. “Knowledge management for manufacturing - the product and process database.” *Journal of Manufacturing Technology Management* 15, no. 7 (2004): 575-584.

Balogun, O, Hawisa H,Tannock J,. “Knowledge management for manufacturing: the product and process database.” *Journal of Manufacturing Technology Management* 15, no. 7 (2004): 575–584.

Bandeira, J., Bittencourt I I, Espinheira P, Isotani S,. “FOCA: A Methodology for Ontology Evaluation.” *Applied Ontology* 3 (2016): 1-3.

Bechky, B.A.,. “Sharing Meaning Across Occupational Communities: The Transformation of Understanding on a Production Floor.” *Orgamisation Science* 14, no. 3 (2003): 312-330.

Benjamin, Perakath, Mukul Patki and Richard Mayer. “Using ontologies for simulation modeling.” *In Proceedings of the 38th conference on Winter simulation.* . Monterey, California: : Winter Simulation Conference, 2006.

Berre, A., B. Elvester, N. Figay, C. Guglielmina, S. Johnsen, D. Karlsen, T. Knothe and S. Lippe. “The ATHENA Interoperability Framework.".” *In 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007).* Madeira, Portugal: Springer, 2007.

Bless, Patrick, N. , Diego Klabjan and Soo Chang, Y. “Heuristics for automated knowledge source integration and service composition.” *Computers and Operations Research* 35, no. 4 (2008): 1292-1314.

Blomqvist, E, A.O¨hgren. “Constructing an enterprise ontology for an automotive supplier.” *Engineering Applications of Artificial Intelligence*, no. 21 (2008): 386-397.

Blomqvist, E., and A. Öhgren. “Constructing an Enterprise Ontology for an Automotive Supplier.” *Engineering Applications of Artificial Intelligence* 21, no. 3 (2008): 386–397.

Bloodsworth, Peter and Sue Greenwood. “COSMOA an ontology-centric multi-agent system for co-ordinating medical responses to large-scale disasters.” *AI Commun* 18, no. 3 (2005): 229-240.

Borgo, S., and P. Leitão. “Foundations for a Core Ontology of Manufacturing.” Portugal: Bragança, 2007.

Borst, W., N.,. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* Tweenty: In Centre for Telematica and Information Technology: University of Tweenty, 1997.

Bourey, J.P., Grangel, R., Doumeingts, G. and Berre, A.,. “INTEROP NoE - Deliverable DTG2.2 - Report on Model Interoperability.” 2006.

Bourey, J.-P.,. “Model Driven Interoperability and Service-Oriented Architecture.” Zurich: PowerPoint Presentation, 2007.

Brank., J, Grobelnik M, Mladenic,. “A survey of ontology evaluation techniques.” 2005.

Brickley D, Guha RV. “RDF Vocabulary Description Language 1.0: RDF.” *W3C Working Draft* (http://www.w3.org/TR/PR-rdf-schema), 2004.

Briefing-paper. *Identifier interoperability.* Digital perspective Europe, 2008.

Brinkley, J. F., Suciu, D., Detwiler, L. T., Gennari, J. H., and Rosse, C. “A framework for using reference ontologies as a foundation for the semantic web.” *AMIA 2006 Symposium Proceedings Page.* 2006.

Brunetti, G. and Golob, B.,. “feature-based approach towards an integrated product model including conceptual design information.” *Computer Aided Design.* 32, no. 14 (2000): 877-887.

Brunnermeier, S.B. and S.A. Martin. *Interoperability Cost Analysis of the U.S. Automotive Supply Chain.* U.S.A: National Institute of Standards and Technology, 2002.

Burgun, A. “Desiderata for domain reference ontologies in biomedicine.” *Journal of Biomedical Informatics* 39 (2006): 307-313.

Bussler, Christoph, Val Tannen, Irini Fundulaki, Philip Tan, Stuart Madnick, Kian-Lee Tan. "Context Mediation in the Semantic Web: Handling OWL Ontology and Data Disparity Through Context Interchange." In *In Semantic Web and Databases*. Berlin / Heidelberg: Springer, 2005.

Castano, S., Antonellis, V.D. "Semantic Dictionary Design for Database Interoperability." *13th International Conference on Data Engineering.* 1997.

Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. "What are ontologies, and why do we need them?" *Intelligent Systems and their Applications,IEEE* 14, no. 1 (1999): 20-26.

Chang, X R. Rai, and J. Terpenny. "Development and Utilization of Ontologies in Design for Manufacturing." *Journal of Mechanical Design* 2, no. 132 (2010).

Chen, D., Vernadet F. "Standards on enterprise integration and engineering - state of the art." *International Journal of Computer Integrated Manufacturing* 178, no. 3 (2004): 235-253.

Chen, D., Doumeingts G., "European inititiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap." *Annual Reveiews in Control* 27, no. 2 (2003): 153-162.

Chen, D., Doumeingts G., "The GRAI-GIM reference model, architecture and methodology." In *Architectures for enterprise integration.*, by P. et al, Bernus. London: Chapman and Hall, 1996.

Chen, D., Doumeingts, G., Vernadat, F.,. "Architectures for enterprise integration and interoperability: past, present and future." *Computers in Industry* 59 (2008): 647-659.

Chen, Yuh-Jen., Yuh-Min Chen., Hui-Chuan Chu., "Development of a mechanism for ontology-based product lifecycle knowledge integration." *Expert Syst. Appl.* 36, no. 2 (2009): 2759-2779.

Chen, Yuh-Min, Ching-Ling Wei., "Computer-aided feature-based design for net shape manufacturing." *Computer Integrated Manufacturing Systems* 10, no. 2 (1997): 147-164.

Chen, David, Verndat Franc, "Standards on enterprise integration and engineering—state of the art." *Journal of Computer Integrated Manufacturing* 17, no. April-May (2004): 235-253.

Chin, K.S., Zhao, Y., Mok, C.K.,. "STEP based multi-view integrated product modelling for concurrent engineering." *International Journal of Advanced Manufacturing Technology.*, 2002: 896-906.

Chungoora, N. *A Framework to Support Semantic Interoperability in Product Design and Manufacture.* Loughborough: Loughboroush University, 2010.

Chungoora, N., Young, R.I.M.,. "Semantic interoperability requirements for manufacturing knowledge sharing." In *Enterprise interoperability III: new challenges and industrial approaches.*, by K., Ruggaber, R., Popplewell, K. and Xu, X., eds. Mertins, 411-422. London: Springer-Verlag, 2008.

Chungoora, N., Young R.I.M. "The configuration of design and manufacture knowledge models from a heavyweight ontologial foundation." *International Journal of Production Research* 49, no. 15 (2011): 4701–4725.

Chungooraa, N., A.-F. Cutting-Decelle, R.I.M. Young, G. Gunendran, Z. Usman,. "Towards the ontology-based consolidation of production-centric standards." *International Journal of Production Research* 2, no. 51 (2013): 327–345.

Ciocoiu, M., Nau, S. D., and Gruninger, M. "Ontologies for Integrating Engineering Applications." *Transactions of the ASME* 1 (2001): 12-22.

Cochrane, Sean, Robert Young, Keith Case, Jennifer Harding, James Gao, Shilpa Dani, David Baxter. “Knowledge reuse in manufacturability analysis.” *Robotics and Computer-Integrated Manufacturing* 24, no. 4 (2008): 508-513.

Cochrane, S D, Case K, Young R I, Harding J A, Dani S. “Knowledge Sharing between Design and Manufacture. I.” *n R.J. Rajiv Khosla, Knowledge-based intelligent information and engineering systems.* berlin: Springer-Verlag, 2005. 221-227.

Cohen, W., P. Ravikumar and S. Fienberg. “A comparison of string distance metrics for name-matching tasks.” *In Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03).* Acapulco, Mexico, 2003.

Collins, C. J. and K. G. Smith. “Knowledge exchange and combination: the role of human resource practices in the performance of high-technology firms.".” *Academy of Management Journal* 49, no. 3 (2006): 544-560.

Conen, W, Klapsing R. *Logical Interpretations of RDFS - A Compatibility Guide. Working paper.* http://nestroy.wi-inf.uni-essen.de/rdf/new_interpretation/, 2001.

Connor, O'Martin, Amar Das. “SQWRL: A Query Language for OWL.” Chantilly, VA: Proceeding OWLED'09 Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529, 2009.

Corcho, O., Fernandez-Lopez, M., and Gomez-Perez, A. “Methodologies, tools and languages for building ontologies. Where is their meeting point?” *Data and Knowledge Engineering* 46 (2002): 41-64.

Corcho, O M, Fernandez-Lopez A. Gomez-Perez. “Methodologies, Tools and Languages for Building Ontologies: Where is their Meeting Point.” *Elsevier Science Publishers BV* 46 (2003): 41–64.

Costa, C A. “Product ontology supporting information exchanging in global furniture industry.” In *Enterprise interoperability II – New challenges and approaches*, edited by R. Gonc¸alves, 278–280. London: Springer, 2007.

Court, A. W. “Issues for integrating knowledge in new product development: reflections from an empirical study.” *Knowledge-Based Systems* 11, no. 7-8 (1998): 391-398.

Cummings, Jonathon, N. “Work Groups, Structural Diversity, and Knowledge Sharing in a Global Organization.” *Management Science* 50, no. 3 (2004): 352-364.

Cutting-Decelle, A.F., Bourey, J.-P., Grangel, R. and Young, R.I.M.,. “Ontology based communications through model driven tools - the MDA approach feasibility of the approach in urban engineering projects.” *In: COST C21 1st Workshop on Ontologies for Urban Development – Interfacing Urban Information Systems.* Geneva, Switzerland., 2006.

Cutting-Decelle, A.F., Dubois, A.M. and Dubois, J.E.,. “An information system for the building industries: a communication approach based on industrial components.” *Data Science Journal* 1, no. 2 (2002): 257-270.

Dartigues, Christel, Parisa Ghodous, Michael Gruninger, Denis Pallez and Ram Sriram. “CAD/CAPP Integration using Feature Ontology.” *Concurrent Engineering*, 2007: 237-249.

Das, B., Cutting-Decelle, A.F., Young, R.I.M., Case, K., Rahimifard, S., Anumba, C.J. and Bouchlaghem, N.,. “Towards the understanding of the requirements of a communication language to support process interoperation in cross-disciplinary supply chains.” *International Journal of Computer Integrated Manufacturing.* (20) 4 (2007): 396-410.

Davenport, Thomas H. “Saving IT's Soul: Human Centered Information Management. Harvard Business Review.” 1994.

Davenport, Thomas H., W. De-Long and Micheal C. Beers. *Successful Knowledge Management Projects.In The Knowledge Management YearBook.* Elsevier, 2000.

Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Broekstra, J., Horrocks, I. “The Semantic Web: The Roles of XML and RDF.” *IEEE Internet Computing,* 4, no. 5 (2000): 63-74.

Delugach, H. S. “Representing metadata constraints in Common Logic.” *Int. J. Metadata, Semantics and Ontologies* 4, no. 4 (2009): 277-286.

Deshayes, Laurent, Sebti Foufou, Michael Gruninger,. “An Ontology Architecture for Standards Integration and Conformance in Manufacturing.” In *Advances in Integrated Design and Manufacturing in Mechanical Engineering II,* by M. Tollenaere and P. Ray S. Tichkiewitch, 261-276. Springer, 2007.

Didonet del Fabro, M., Albert, P., Bézevin, J. and Jouault, F.,. *Industrial-strength rule interoperability using model driven engineering.* Rennes.: Research Report Version 1. France: Institut de Recherche en Informatique et en Automatique,, 2008.

Dinar M, Rosen D,. “A Design For Additive Manufacturing Ontology.” *Interntional Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Charlotte, North Carolina: ASME, 2016.

Elanchezhian, C., Selwyn, T. S., and Sundar, G. S. *Computer Aided Manufacturing.* First ed. New Delhi:: Laxmi Publications, 2005.

Elvesæter, B., Hahn, A., Berre, A.-J. and Neple, T.,. “Towards an interoperability framework for model-driven development of software systems.” In *Interoperability of enterprise software and applications,* by D., Bourrières, J.-P., Léonard, M. and Boudjilida, N., eds. Konstantas, 409-420. London,UK: Springer-Verlag., 2006.

Farhad Ameri, Christian McArthur. “Semantic rule modelling for intelligent supplier discovery.” *International Journal of Computer Integrated Manufacturing* 27, no. 6 (2015): 570-590.

Fathi, M.N., Eze, U.C., Goh, G.G. “Key determinants of knowledge sharing in an electronics manufacturing firm in Malaysia.” *Library Review* 60, no. 1 (2011): 53-67.

Feng, S.C. and Song, E.Y.,. “A manufacturing process information model for design and process planning integration.” *Journal of Manufacturing Systems.* 22, no. 1 (2003).

Fenves, Steven J. , Sebti Foufou, Conrad Bock and Ram D. Sriram. “CPM: A Core Model for Product Data.” 2006.

Fernández-López, M. “Overview of Methodologies for Building Ontologies.” Stockholm.: IJCAI99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, 1999.

Fernández-López, M. and Gómez-Pérez, A. “Overviews and analysis of methodologies for building ontologies.” *The Knowledge Engineering Review* 17, no. 2 (2002): 129-156.

Fernández-López, M. F., A. Gomez-Perez, J. P. Sierra and A. P. Sierra. “Building a chemical ontology using Methontology and the Ontology Design Environment.” *Intelligent Systems and their Applications, IEEE* 14, no. 1 (1999): 37-46.

Fernandez-Lopez, Mariano., Asuncion. Gomez-Perez and N. Juristo. “METHONTOLOGY: From Ontological Art Towards Ontological Engineering.” *In AAAI Symposium on Ontological Engineering.* Stanford, 1997.

FinES-Cluster. “2012 Fines Standardization TF.” 2011. http://www.fines-cluster.eu/fines/wp/standardizationtf/2012/03/27/7-4-1-foundation-and-domain-ontologies/#7.

Firat, A., Madnick, S., and Grosof, B.,. “Contextual alignment of ontologies in the eCOIN semantic interoperability framework.” *Information Technology Management.* 8 (2007): 47-63.

Fischer, Ulli and Dragan Stokic. *Organisational Knowledge Management in Manufacturing Enterprises-Solutions and Open Issues.* The project PICK, 2002.

Fowler, J.,. *STEP for data management, exchange and sharing.* Great Britain:: Technology Appraisals, 1996.

Frankovič, Baltazár., Ivana Budinská. “The Role of Ontology in Building of Knowledge Systems for Industrial Applications.” *4th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence.* Herlany, Slovakia, 2006.

Frappaolo., Carl,. *Knowledge Management.* Capstone Publishing Ltd., 2006.

Fürst, F. “Axiom-Based Ontology Matching.” New York, NY, USA: Proceedings of the 3rd international conference on Knowledge capture, 2005. 195-196.

Gallaghar, C.C. and W.A. Knight. *Group technology production methods in manufacture.* Chichesterm UK: Ellis Horwood, 1986.

Gallaher, Michael P, Alan C O‘Connor, John L Dettbarn, Jr. and Linda T Gilday. *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry.* Gaithersburg, Maryland: U.S. Department of Commerce Technology Administration, National Institute of Standards and Technology (NIST), 2009.

Gangemi, Aldo and Stefano Borgo. “Core Ontologies in Ontology Engineering 2004 (Un)Successful cases and best practices for ontology engineering: reusing well-founded ontologies for domain content specification.” *In Proceedings of the EKAW*04 Workshop on Core Ontologies in Ontology Engineering.* Northamptonshire (UK)., 2004.

Gasevic, D., Djuric, D., and Devedzic, V. *Model Driven Architecture and Ontology Development.* Berlin: Springer-Verlag, 2006.

Giachetti, R.E.,. “A standard manufacturing information model to support design for manufacturing in virtual enterprises.” *Journal of Intelligent Manufacturing.* 10 (1999): 49-60.

Gnägi, H.R., Morf, A. and Staub, P.,. “Semantic interoperability through the definition of conceptual model transformations.” *In: Proceedings of the 9th AGILE Conference on Geographic Information Science.* Visegrád, Hungary, 2006.

Go´mez-Pe´rez, A., Ferna´ndez-Lo´pez, M., and Corcho, O. *Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic web.* London: Springer, 2004.

Gómez-Pérez., Asunción. "Evaluation of ontologies." *International Journal of intelligent systems* 16 (2001): 391-409.

Grenon, P. "[DL] CFP: Reference Ontology and Applications Ontology: ." *Workshop in Hamburg, 15-18 September 2003.* Hamburg: http://dl.kr.org/pipermail/dl/2003/001594.html, 2003.

Gruber, T. R.,. "A translation approach to portable ontology specification." *Knowledge Acquisition* (5) 2 (1993): 199-220.

Gruber., T R,. *Toward principles for the design of ontologies used for knowledge sharing.* Academic Press, Inc, 1995.

Gruninger, M. and J. Lee. "Introduction: Ontology Applications and Design." *Communications of the ACM* 45, no. 2 (2002).

Gruninger, M., and M. Fox. "Methodology for the Design and Evaluation of Ontologies." Montreal, Canada: IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.

Gruninger, M., Atefi, K., and Fox, M. S. "Ontologies to Support Process Integration in Enterprise Engineering." *Computational and Mathematical Organization Theory* 6 (2001): 381-394.

Gruninger., M, Uschold M,. "Ontologies:Principles methods and applications." *The Knowledge Engineering Review* 11, no. 2 (1996): 93-155.

Gu, P. "A feature representation scheme for supporting integrated manufacturing." *Computers and Industrial Engineering* 26, no. 1 (1994): 55-71.

Guarino, N. "Understanding, building and using ontologies." *International Journal Human-Computer Studies* 46 (1997): 293-310.

Guarino, N., and Giaretta, P. "Ontologies and Knowledge Bases Towards a Terminological Clarification." In *Towards Very Large Knowledge Bases*, by N. Mars, 25-32. Amsterdam: IOS Press, 1995.

Guarino, N., Oberle, D., Staab, S.,. "What is Ontology?" In *Handbook on Ontologies*, by S., Studer, R., Staab, 1-21. London: Springer, 2009.

GUIDE, MDA. *MDA Guide. Version 1.0.1.* . Document Number: omg/2003-06-01., 2003.

Gunendran, A.G., Young, RIM., Cutting-Decelle A.F., Bourney, J.P., "Organising manufacturing information for engineering interoperability. In: R. Gonc¸alves, et al." In *Enterprise interoperability II – New challenges and approaches.*, 587-598. London: Springer, 2007.

Gunendran, A. G. and R. I. M. Young. "Methods for the capture of manufacture best practice in product lifecycle management." *International Journal of Production Research* 48, no. 20 (2010): 5885-5904.

Gunendran, A.G. and Young, R.I.M.,. "An information and knowledge framework for multi-perspective design and manufacture." *International Journal of Computer Integrated Manufacturing.* 19, no. 4 (2006): 326-338.

Gunendran, George and Bob Young. “Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management.” *International Conference on Product Lifecycle Management 2008.* inderscince publishers, 2008.

Gunendran., A G. *An Information and Knowledge Framework to Support Multiple Viewpoints in the Design for Manufacture Of Injection Moulded Products.* Louhborough: Loughborough University, 2004.

Gupta, R.K. and Gurumoorthy, B.,. “A feature-based framework for semantic interoperability of product models.” *Journal of Mechanical Engineering* 54, no. 6 (2008): 446-457.

Hakimpour, Farshad. *Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata.* Zurich: University of Zurich, 2003.

Han, J.H. and A.A.G. Requicha. “Integration of Feature Based Design and Feature Recognition.” *Proceedings of ASME 15th International Conference in Engineering Conference.* Boston, MA, USA, 1995.

Hansen, Morten T. “Knowledge Networks: Explaining Effective Knowledge Sharing in Multiunit Companies.” *Organization Science* 13, no. 3 (2002): 232-248.

Harding, J. A. *A knowledge representation model to support concurrent engineering teamwork." In Mechanical and Manufacturing Engineering.* Loughborough: Loughborough University, 1996.

Harding, J. A. and B. Yu. *Information-centred enterprise design supported by a factory data model and data warehousing.* Elsevier Science Publishers B. V., 1999.

Hardwick, M., Spooner, D. L., Rando, T., Morris, K.C. “Sharing Manufacturing Information in Virtual Enterprises.” *Communications of the ACM* 39, no. 2 (1996): 46-54.

Hastilow, N. *Manufacturing Systems Interoperability in Dynamic Chnage Environments. PhD Thesis.* Loughborough: Loughborough University, 2013.

He., Y, Chuanpeng Hao, Yulin Wang b, Yufeng Li , Yan Wang, Lingyu Huang. “An ontology-based method of knowledge modelling for remanufacturing process planning.” *Journal of Cleaner Production* 258 (2020): 120952.

Heijst., G V, Schreiber A T,Wielinga B J,. “Using explicit ontologies in KBS development.” *International Journal of Human-Computer Studies* 46, no. 2-3 (1996): 183-292.

Hendriks, Paul. “Why share knowledge? The influence of ICT on the motivation for knowledge sharing.” *Knowledge and Process Management* 6, no. 2 (1999): 91-100.

Hlomani, H. and STACEY, D. “Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey.” *Semantic Web Journal*, 2014: 1-5.

Hoof, B. v.-d., Ridder, J.A. “Knowledge Sharing in context:the infuence of organisational commitment, communication climate and CMC use of knowldge sharing.” *Journal of Knowledge Management* 8, no. 6 (2004): 117-130.

Houtzeel, A. “MICLASS-A classification system based on group technology.” *SME Technical Paper:MS*, 1975: 75-79.

Huang, T.-T., Chen, L., Stewart, R. A. “The moderating effect of knowledge sharing on the relationship between manufacturing activities and business performance.” *Knowledge Management Research and Practise* 8 (2010): 285-306.

Huang., N, Diao S,. “Ontology-based enterprise knowledge integration.” *Robotics and Computer Integrated Manufacturing* 24, no. 4 (2008): 562-571.

IDEAS. *Project Deliverables (WP1-WP7), Public Reports,*. www.ideas-road map.net, 2003.

IDEF5 Method Report. *Integration DEfiniton for Function modelling (IDEF5).* Prepared for:Armstrong Laboratory AL/HRGAWright-Patterson Air Force Base, Ohio, Knowledge Based Systems, Inc., 1994.

IEEE-Std-Computer-Dictionary, .,. *IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries.* In IEEE Std 610, 1991.

Imran, I. *Towards an Assembly Reference Ontology for Assembly Knowldge Sharing. PhD Thesis.* Loughborough: Loughborough University, 2013.

Imran, M and Young Bob. “The application of common logic based formal ontologies to assembly knowledge sharing.” *Journal of Intelligent Manufacturing*, 2013.

International Standard, First edition 2007-10-01 Information technology — Common Logic (CL): a framework for a family of logic based languages.". “Representation of Business Rules in UML and OCL Models for Developing Information Models.” *In A. P. Janis Stirna, The Practice of Enterprise Modeling: First IFIP WG 8.1 Working Conference, PoEM 2008.* Laxenburg: IFIP International Federation For Information Processing, 2009.

ISO 18629-1. “Industrial Automation System and Integration—Process Specification Language: Part 1. Overview and Basic Principles.” 2004.

ISO 19115. *ISO Online Browsing Platform (OBP).* 2003. https://www.iso.org/obp/ui/#search (accessed 2018).

ISO/CEN-11354. “Requirements for establishing manufacturing enterprise process interoperability: Part 1: Framework for Enterprise Interoperability.” 2008.

ISO/IEC 10746-3. “Information technology - open distributed processing - reference model: architecture.” 1996.

ISO/IEC-24707. “International Standard, First edition 2007-10-01 Information technology — Common Logic (CL): a framework for a family of logic based languages."." 2007.

ISO/IEC-TR-10000-3. *Information technology -- Framework and taxonomy of International Standardized Profiles -- Part 3: Principles and Taxonomy for Open System Environment Profiles,*. ISO, 2008.

ISO/IEC-TR-10000-3. *Information technology -- Framework and taxonomy of International Standardized Profiles -- Part 3: Principles and Taxonomy for Open System Environment Profiles.* ISO, 1998.

ISO/TC44. *International Standardisation Organiation Technical Committee 44.* 2017. https://committee.iso.org/home/tc44 (accessed October 2017).

ISO/TS-10303-1164. *Industrial automation systems and integration -- Product data representation and exchange -- Part 1164: Application module: Product as Individual.".* ISO, 2004.

ISO-10303-1. *Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 1: Overview and Fundamental Principles.* International Organisation of Standarads, 1994.

ISO-13584. *Industrial automation systems and integration – Parts library.* International Organisation of Standards., 2001.

ISO-16100-1. *Industrial automation systems and integration -- Manufacturing software capability profiling for interoperability -- Part 1: Framework.* International Organisation of Standards, 2009.

Jagenberg, Jan Tim, Erik A. Gilsdorf, Reiner Anderl and Thomas Bornkessel. "Knowledge Driven Design Features for the Product Life Cycle of Engine Parts." *Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2009.* San Diego, California, USA., 2009.

Jan de Kraker, K. *Feature Conversion for Concurrent Enginnering.* Delft, The Netherlands: Computer Science. Delft University of Technology, 1998.

Jean, S., Pierra, G., and Ait-Ameur, Y. *Domain Ontologies : A Database-Oriented Analysis.* Setubal, Portugal: Proc. of Web Information Systems and Technologies., 2006.

Julio Cesar Nardi, Ricardo de Almeida Falbo, JoãoPauloA.Almeida, Giancarlo Guizzardi, MartenJ.van Sinderen, Nicola Guarino, Claudenir Morais Fonseca. "A commitment-based referenceontology for services." *Information Systems* 54 (2015): 263–288.

Jung, J.Y. and R.S.. Ahluwalia. "A coding and classification system for formed parts." *Journal of Manufacturing Systems* 10, no. 3 (1991): 223-232.

KACTUS. *The KACTUS Booklet version 1.0. Esprit Project 8145 KACTUS.* http://www.swi.psy.uva.nl/projects/NewKACTUS/Reports.html, 1996.

Khondoker, R. M., and Mueller, P. "Comparing Ontology Development Tools Based on an Online Survey." *Proceedings of the World Congress on Engineering.* London: WCE, 2010.

Kim., K Y, Manley D G, Yang H,. "Ontology-based assembly design and information sharing for collaborative." *Computer-Aided Design* 38, no. 12 (2006): 1233–1250.

Kiryakov, A.K., Simov, K. Iv. and Dimitrov, M.,. "OntoMap: ontologies for lexical semantics." *In: Proceedings of the Euro Conference Recent Advances on Natural Language Processing.* Bulgaria, 2001. 142-148.

Kleppe, A., Warmer, J.,Bast, W. *MDA EXPLAINED The Model Driven Architecture: Practice and Promise.* PAPERBACK., 2003.

Komatsoulis, G. A., Warzel, D. B., Hartel, F. W., Shanbhag, K., Chilukuri, R., Fragoso, G.,. "caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability." *Journal of Biomedical Informatics,* 41 (2008): 106-123.

Kosanke, K., F. Vernadat , M. Zelm. "CIMOSA: enterprise engineering and integration." *Computers in Industry* 40, no. 2-3 (1999): 83-97.

Kuffner, T.A., Ullman, D.G. "The information requests of mechanical design engineers." *Design Studies* 12, no. 1 (1991): 42-50.

Kuffner, Tom A. and David G. Ullman. “The information requests of mechanical design engineers.” *Design Studies* 12, no. 1 (1991): 42-50.

Lambert, D., Saulwick, A., Nowak, C., Oxenhan, M., O'Dea, D. *An Overview of Conceptual Frameworks.* Edinburgh: Commonwealth of Australia: DSTO Defence Science and Technology Organisation, Command, Control, Communications and Intelligence Division, Department of Defence., 2009.

Lassila, O., Swick, R. R. *Resource Description Framework (RDF) Model and Syntax Specification.* http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, 1999.

Lazenberger, M., Sampson, J.J., Rester, M., Naudet, Y. and Latour, T.,. “Visual ontology alignment for knowledge sharing and reuse.” *Journal of Knowledge Management.* 12, no. 6 (2008): 102-120.

Lee, G., Eastman, C.M., Sacks, R. and Navathe, S.B.,. “Grammatical rules for specifying information for automated product data modelling.” *Advanced Engineering Informatics* 20 (2006): 155-170.

Lee, Jaehyun, Hyowon Suh and Soon Hung Han. “Ontology-based Knowledge Framework for Product Development.” *Computer Aided Design and Applications* 2, no. 5 (2005): 635-643.

Lee, Jae-Nam. “The impact of knowledge sharing, organizational capability and partnership quality on IS outsourcing success.” *Information and Management* (38) 5 (2001): 323-335.

Leila, Z.-G. *Reference Ontology Presentation.* http://www.slideshare.net/ontoini/sitis-kare-09-reference-ontology-presentation, 2009.

Leimagnan, Severin, Ali Siadat, Jean-Yves Dantan and Anatoli Semenenko. “MASON: A proposal for an ontology for manufacturing domain.” *In Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS‘06).* IEEEXplore, 2006.

Lemrabet, Youness, Michel Bigand, David Clin, Nordine Benkeltoum and Jean-Pierre Bourey. “Model driven interoperability in practice: preliminary evidences and issues from an industrial project.” *In Proceedings of the First International Workshop on Model-Driven Interoperability.* Oslo, Norway: ACM, 2010.

Lenat, D., B., and R. Guha, V.,. “Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.” *Artificial Intelligence*, 1990.

Li, Z., Raskin, V. and Ramani, K.,. *A methodology of Engineering Ontology Development for Information Retreaval.* West Lafayette: Purdue University, 2007.

Liang., J S,. “An ontology-oriented knowledge methodology for process planning in additive layer manufacturing.” *Robotics and Computer Integrated Manufacturing* 53 (2018): 28-44.

Lin, H K, Harding J A,. “A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration.” *Computers in Industry* 58 (2007): 428–437.

Lin, Hsiu-Fen. “Knowledge sharing and firm innovation capability: an empirical study.” *International Journal of Manpower* 28, no. 3/4 (2007): 315 - 332.

Lin., H K, Harding J A,. “A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration.” *Computers in Industry* 58 (2007): 428–437.

Liping, Zheng, Li Guangyao, Liang Yongquan and Sha Jing. “Design of ontology mapping framework and improvement of similarity computation.” *Journal of Systems Engineering and Electronics* 18, no. 3 (2007): 641-645.

Liu, S. and Young, R.I.M.,. “Utilising information and knowledge models to support global manufacturing coordination decisions.” *International Journal of Computer Integrated Manufacturing* 17, no. 4 (2004): 479–492.

Lopez-Ortega., O.,Ramirez M,. “A STEP-based manufacuring information system to share flexible manufacturing resources data.” *Journal of Intelligent Manufacturing* 16 (2005): 287-301.

Love, D.M. “Flexibility in the design and application of component coding and classification software.” *National Conference on Production Research.* Nottingham, UK., 1985.

Lyons, J.,. *Semantics 1.* Cambridge: Great Britain: Cambridge University Press., 1977.

Ma, Y.-S., Britton, G.A., Tor, S.B. and Jin, L.Y. “Associative assembly design features - concept, implementation and application.” *International Journal of Advanced Manufacturing Technology* 32 (2007): 434-444.

Maedche, A. *Ontology Learning for the Semantic Web.* Norwell: Kluwer Academic Publishers., 2003.

Manufacture, A Vision for 2020. *Manufacture, A Vision for 2020, Assuring the Future of Manufacturing in Europe.* European Commission Report of the High-level Group, 2004.

Manupati., V K, Putnik G D, Tiwari M K, Ávila P, Cruz-Cunha M M,. “Integration of process planning and scheduling using mobile-agent based approach in a networked manufacturing environment.” *Computers and Industrial Engineering* 94 (2016): 63-73.

Martino, T.D. , Giannini, F.,. “Feature-based product modelling in concurrent engineering.” *Proceedings of the 10th International Conference on Information Technology in the Product Realisation Process.* Trento, Italy., 1998.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A.,. *WonderWeb Deliverable D18 - Ontology Library.* Trento, Italy.: Laboratory for Applied Ontology - ISTC-CNR., 2003.

McGuinness, Natalya , F. Noy and L. Deborah. “Ontology Development 101: A Guide to Creating Your First Ontology.” Stanford, CA.: Stanford University, 2002.

McInerney, Claire, Koenig, Michael E. D.,. *Knowledge Management (KM) Processes in Organizations: Theoretical Foundations and Practice.* Morgan and Claypoo, 2011.

Mertins, K., Ruggaber, R., Popplewell, K., Xu, X. *Preface. In: Mertins, K., et al eds. Enterprise Interoperability III:New Challenges and Industrial Approches.* London: Springer-Verlag, 2008.

Mesmer-Magnus, Jessica R. and Leslie A. DeChurch. “Information Sharing and Team Performance: A Meta-Analysis.” *Journal of Applied Psychology* 94, no. 2 (2009): 535-546.

Michel, J. “Terminology extracted from some manufacturing and modelling related standards.” CEN/TC 310 N1119R2, 2005.

Mizoguchi, R. “Tutorial on Ontological Engineering, Part 2: Ontology Development, Tools and Languages.” *New Generation Computing,* 22 (2004): 61-96.

Mizoguchi, R., and Kozaki, K. “Ontology Engineering Environments.” In *Handbook on Ontologies*, by and R. Studder S. Staab, 315-335. Berlin: Springer-Verlag, 2009.

Moalla, N., Ouzrout, Y., Neubert, G. and Bouras, A.,. “Model-driven interoperability to enhance product data quality.” *In: Proceedings of the 1st International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS’08) in Conjunction with the CAiSE’08 Conference.*

Montpellier, France: [Online] Available at: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-340/, 2008. 121-133.

Molina, A., Bell R.,. "A Manufacturing Model representation of a flexible manufacturing facility." *Journal of Engineering Manufacture: Proceedings of the Institution of Mechanical Engineers 213, Part B*, 1999: 225-246.

Molina, A., Ellis, T.I.A., Young, R.I.M. and Bell, R.,. "Modelling manufacturing capability to support concurrent engineering." *Concurrent Engineering Research and Applications.* 3, no. 1 (1995): 29-42.

Musen, M. A. "Dimensions of Knowledge Sharing and Reuse." *Computers and Biomedical Research* 25 (1992): 135-467.

Musen, M. A. *Domain Ontologies in Software Engineering: Use of Protégé with the EON Architecture.* California: Stanford, 1998.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. "Enabling Technology for KNowledge Sharing." *AI Magazine* 12, no. 3 (1991): 37-56.

Niles, I., Pease, A. *Towards a Standard Upper Ontology.* Palo Alto, CA: Teknowledge Corporation, 2001.

Nitishal Chungoora, George A Gunendran, Robert IM Young, Zahid Usman, Najam A Anjum, Claire Palmer, Jenny A Harding, Keith Case, Anne-Françoise Cutting-Decelle. "Extending product lifecycle management for manufacturing knowledge sharing." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* (Sage Publications) 226, no. 12 (2012): 2047-2063.

Nonaka, Ikojiro, Hirotaka Takeuchi. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation.* Oxford: Oxford University Press, 1995.

Noy, N.F., Musen, M.A.,. "The PROMPT suite: interactive tools for ontology merging and mapping." *International Journal of Human-Computer Studies* 59 (2003): 983-1024.

Noy, Natalya F. "Tools for Mapping and Merging Ontologies." In *In Handbook on Ontologies*, by Steffen Staab and Rudi Studer. Birkhäuser.: Springer, 2004.

Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M.,Zhou, J. "DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO)." *Journal of Web Semantics* 5, no. doi:10.1016/j.websem.2007.06.002 (2007): 156-174.

Oberle, D., Grimm, S., Staab, S. "An Ontology for Software." In *Handbook on Ontologies, International Handbooks on Information Systems*, by Staab Stuber. Berlin: Springer-Verlag, 2009.

Oberle, Daniel., "Semantic management of middleware." *In Proceedings of the 1st international doctoral symposium on Middleware.* Toronto, Ontario, Canada: ACM, 2004.

Obrst., L, Werner C, Inderjeet M, Steve R,Smith B,. "The Evaluation of Ontologies: Toward Improved Semantic Interoperability." In *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, edited by Baker, and K.-H. Cheung Dans JO Christopher. 2007.

Opitz, H. *A classification system to describe workpieces, Part I and II.* Pergamon Press, 1970.

Otto, H.E., “From concepts to consistent object specifications - translation of a domain-oriented feature framework into practice.” *Journal of Computer Science and Technology* 16, no. 3 (2001): 208-230.

Ouksel, A. M., Sheth, S., “Semantic Interoperability in Global Information Systems.” *SIGMOD Rec.* 28, no. 1 (1999): 5-12.

Oxford Dictionary. *Oxford English Dictionary.* 2019. https://www.oed.com/view/Entry/131551?redirectedFrom=ontology#eid.

Oxford. *Oxford Dictionary.* 3rd. Oxford: Oxford University Press, 2010.

Ozman, Müge., “Knowledge integration and network formation.".” *Technological Forecasting and Social Change* 73, no. 9 (2006): 1121-1143.

Oztemal, E., Tekez, E.K., “Integrating manufacturing systems through knowledge exchange protocols within an agent-based knowledg network.” *Robotics and Computer-Integrated Manufacturing* 25 (2009): 235-245.

Palmer, C, Usman Z, Junior O S, Malucelli A, Young RIM,. “Interoperable Manufacturing Knowledge Systems.” *International Journal of Production Research* 56, no. 8 (2018): 2733-2752.

Panetto, H. “Towards a classification framework for Interoperability of Enterprise Applications.” *International Journal of Computer Integrated Manufacturing* 20, no. 8 (2007): 727-740.

Panetto, H., Scannapieco, M., Zelm, M. “INTEROP NoE – interoperability research for networked enterprises applications and software.” *In: Meersman, R. et al, eds. On the Move to Meaningful Internet Systems. OTM 2004 Workshops. LNCS 3292.* Berlin Heidelberg: Springer, 2004. 866-882.

Patil, L., D. Dutta, R. Sriram. *Ontology Formalization of Product Semantics for Product Lifecycle Management. In Manufacturing Systems Integration Division,.* Gaithersburg: National Institute of Standards and Technology (NIST), 2005.

Patil, L., Dutta, D., Sriram, R.,. “Ontology-based exchange of product data semantics.” *IEEE Transactions on Automation Science and Engineering.* 2, no. 3 (2005): 213-225.

PERA. *Preliminary study of geometric-based system of component description.* PERA production system report-207, 1969.

Polovina, S., Cook, J., Loke, J., “A Practical Exploration of Ontology Interoperability. In F. D. Sebastian Rudolph, Conceptual Structures: Leveraging Semantic Technologies.” *17th International Conference on Conceptual Structures, ICS 2009.* Moscow, Russia: Springer-Verlag, 2009. 247-256.

Pouchard, L., Ivezic, N.,Schlenoff, C.,. “Ontology engineering for distributed collaboration in manufacturing.” *In: Proceedings of the AIS2000 Artificial Intelligence and Simulation Conference.* Tucson, Arizona, USA, 2000.

Pratt, M. J. *Introduction to ISO 10303 - the STEP Standard for Product Data Exchange. TECHNICAL NOTE.* NIST, 2001.

Pratt, M. J.,Wilson, P. R. “Requirements for support of form features in a solid modelling system.” *Computer Aided Manufacturing-International*, 1985.

Ray, S.R., Jones, A.T., “Manufacturing interoperability.” *Concurrent Engineering: Enhanced Interoperable System." In 10 th ISPE international Conference.* Madeira Island - Portugal, 2003.

Ray, S.R. *Semantic approach to standards and interoperability. PowerPoint presentation.* National Institute of Stanbdards and Technology (NIST)., 2004.

Ray, S.R., Jones, A.T. "Manufacturing Interoperability." *Journal of Intelligent Manufacturing* 17 (2006): 681-688.

Riege, A. "Three-dozen knowldge sharing barriers managers must consider." *Journal of Knowledge Management* 9, no. 3 (2005): 18-35.

Roche, C.,. "Corporate ontologies and concurrent engineering." *Journal of Materials Processing Technology,* no. 107 (2000): 187-193.

Roche, Christophe. "Corporate ontologies and concurrent engineering." *Journal of Materials Processing Technology* 107, no. 1-3 (2000): 187-193.

Roller, D. "Design by Features: An Approach to High Level Shape Manipulation." *Computers in Industry* 12, no. 3 (1989): 185-191.

Rosen, D. W. "Feature-Based Design: Four Hypotheses for Future CAD Systems." *Research in Engineering Design* 5 (1993): 125-139.

Ruggaber, R. "ATHENA – Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications." In *In Interoperability of enterprise software and applications (IESA),,* by Bourrières, J.-P., Léonard, M., Boudjilida, N. D. Konstantas. London, UK: Springer-Verlag, 2006.

Rumbaugh J, Jacobson I, Booch G. *The Unified Modeling Language.* Boston, Massachusetts: Addison-Wesley, 1998.

Rynn, Jon. *Six Reasons Manufacturing is Central to the Economy.* New York: Roosevelt Institute, 2011.

Saaksvuori, A., and Immonen, A. *Product Lifecycle Management.* Berlin Heidelberg: Springer-Verlag, 2008.

Saha. S, Usman Z, Li Wei, Jones S, Kshirsagar R,. "A Formal Ontological Approach for Semantic Coherence in Welding Standards." In *Advances in Manufacturing Technology XXXI*, edited by J. Gao et al., 503-508. London: IOS Press, 2017.

Saha. S, Usman Z, Li Wei, Jones S, Shah N,. "Core domain ontology for joining processes to consolidate welding standards." *Robotics and Computer Integrated Manufacturing* 59 (2019): 417-430.

Saha., S, Usman Z, Li Wei, Jones S, Kshirsagar R,. "Towards A Formal Ontology To Support Interoperability Across Multiple Product Lifecycle Domains in the Proceedings of IEEE 11th International Conference on Semantic Computing." San Diego, CA: IEEE Xplore, 2017.

Salomons, O. W., F. J. A. M. van Houten and H. J. J. Kals. "Review of research in feature-based design." *Journal of Manufacturing Systems* 12, no. 2 (1993): 113-132.

Sánchez, Diana Marcela , José María Cavero and Esperanza Marcos Martínez. "The road towards ontologies." In *In Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems,,* by Rajiv Kishore and Ram Ramesh Raj Sharman, 3-21. Springer, 2007.

Sanchez-Alonso, S., and Garcia-Barriocanal, E. "Making use of upper ontologies to foster interoperability between SKOS concept schemes." *Online Information Review,* 30, no. 3 (2006): 253-277.

Sanfilippo, E.M., Borgo S.,. “What are features? An ontology-based review of literature.” *Computer-Aided Design* 80 (2016): 9-8.

Sanya, I O, Shehab E M,. “An ontology framework for developing platform-independent knowldge-based engineering systems in aerospace industry.” *International Journal of Production Research* 52, no. 20 (2014): 6192-6215.

Sarkar., A, Šormaz, D,. “Ontology Model for Process Level Capabilities of Manufacturing Resources.” *Procedia Manufacturing* 39 (2019): 1889 - 1898.

Schlenoff, C. , M. Gruninger, F. Tissot, J. Valois, J. Lubell and J. Lee. *ISO-18629 The Process Specification Language (PSL): Overview and Version 1.0 Specification.* Gaithersburg, MD: NISTIR 6459, National Institute of Standards and Technology, 2000.

Schreiber, G., B. Wielinga, R. de Hoog, H. Akkermans and W. Van de Velde. “CommonKADS: a comprehensive methodology for KBS development.” *IEEE Expert* 9, no. 6 (2000): 28-37.

Schreiber, G., Wielinga, B., and Jansweijer, W. “The Kactus View on the 'O' Word.” *In the proceeding of IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing.* Montreal: Academic Press, 1995.

Scott, Robert E. *The Importance of Manufacturing.* Economic Policy Institute, 2008.

SCRA. *STEP Application Handbook - ISO 10303 Version 3.* 2006. http://www.tc184-sc4.org/.

Semere, Daniel, T, Saqib Dislshad and Bengt Lindberg. “Machining ontology and knowledge modelling.” *In Swedish Production Symposium.* Sweden, 2007.

Sengupta, K., and Hitzler, P. *Web Ontology Language (OWL) Wright State University, Joshi Research Center, Dayton.* 2013. http://knoesis.wright.edu/pascal/pub/owl-encyc-13.pdf (accessed 11 2013).

Shah, J.J. *Parametric and feature-based CAD/CAM - concepts, techniques and applications.* New York, USA: Wiley, 1995.

Simpson, J. A., R. J. Hocken, J. S. Albus. “The automated manufacturing research facility of the national bureau of standards.” *Journal of Manufacturing Systems* 1, no. 1 (1982): 17-32.

Sirin, E., Smith, M., and Wallace, E.,. “Opening, closing worlds – on integrity constraints. In: The 5th international workshop on OWL experiences and directions (OWLED 2008).” Karlsruhe, Germany: OWLED 2008, 2008.

Solano, L, Romero F, Rosado P,. “An ontology for integrated machining and inspection process planning focusing on resource capabilities.” *International Journal of Computer Integrated Manufacturing* 29, no. 1 (2016): 1-15.

Šormaz., Dušan , Arkopaul Sarkar. “SIMPM – Upper-level ontology for manufacturing process plan network.” *Robotics and Computer Integrated Manufacturing* 55 (2019): 183-198.

Spek, Rob van der., André Spijkervet. *Dealing Intelligently With Knowledge.* 2005.

Srinivasan, Vijay. “An integration framework for product lifecycle management.” *Computer-Aided Design* 43 (2011): 464-478.

Staab S, Erdmann M, Maedche A, Decker S. “An Extensible Approach for Modeling Ontologies in RDF(S).” *In: Constantopoulos P (ed) ECDL 2000 Workshop on the Semantic Web,.* Lisbon, Portugal, 2000. 11–22.

Staab, S., R. Studer, H. P. Schnurr and Y. Sure. "Knowledge processes and ontologies." *Intelligent Systems IEEE* 16, no. 1 (2001): 26-34.

Staab, S., Studer, R.,. *Handbook of Ontologies* . London: Springer, 2009.

S-TEN Project, S-TEN. *Intelligent self-describing technical and environmental networks SemanticSTEP [online]*. 2011. http://www.s-ten.net/ (accessed 11 2017).

Studer, R., Benjamins, V. R., and Fensel, D. "Knowledge Engineering: Principles and Methods." *Data and Knowledge Engineering* 25 (1998): 161-197.

Sudarsan, R., Fenves, S.J., Sriram, R.D. and Wang, F., 2005. "A product information modelling framework for Product Lifecycle Management." *Computer Aided Design* 37 (2005): 1399-1411.

Sureephong, P., N. Chakpitak, Y. Ouzrout and A. Bouras. "An Ontology-based Knowledge Management Systemfor Industry Clusters." *International Conference on Advanced Design and Manufacture (ICADAM 2008).* Sanya, China, 2008.

Sveiby, Karl, Erick. *The New Organizational Wealth: Managing and Measuring Knowledge-Based Assets.* San Francisco: Berrett-Koehler Inc, 1997.

Swartout, B., P. Ramesh, K. Knight and T. Russ. "Toward Distributed Use of Large-Scale Ontologies." *AAAI Symposium on Ontological Engineering.* 1997.

TC184/SC4. *Setting the standards for industrial data.* [Online] Available at:http://www.tc184-sc4.org/, 2009.

TOGAF. *The Open Group Architecture Framework.* http://www.togaf.org/, 2009.

Trade-policy-review-report. "Trade policy review report by European communities, World Trade Organization-Trade Policy Review." 2009.

True, M. and C. Izzi. *Collaborative Product Commerce: Creating Value Across the Enterprise.* Accenture, 2002.

Tsui, L., Chapman, S.A., Schnirer, L., Stewart, S. *A Handbook on Knowldge Sharing:Strategies and Recommendations for Researchers, Policymakersand Servie Providers.* 2006.

Tu, Zhiying., Zacharewicz, G., Chen, D. "A federated approach to develop enterprise interoperability." *International Journal of Intelligent Manufacturing* 27, no. 1 (2016): 11-31.

Turban, E. and J. Aronson. 2005. *Decision Support Systems and Intelligent Systems, 7 Edition.* Prentice Hall, Inc, 2005.

Tursi, A., H. Panetto, G. Morel and M. Dassisti. "Ontological approach for products-centric information system interoperability in networked manufacturing enterprises." *Annual Reviews in Control* 33, no. 2 (2009): 238-245.

Uschold, M., and M. King. "Towards a Methodology for Building Ontologies." Montreal: In IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing edited by Chris S. Mellish., 1995.

Uschold, M., Gruninger G,. "Ontologies: principles, methods, and applications." *Knowledge Engineering Review* 11, no. 2 (1996): 93-155.

Uschold, M., Martin King, Stuart Moralee, Yannis Zorgios. "The Enterprise Ontology." *Knowledge Engineering Review* 13 (1998): 31-89.

Uslar, M.,. “Ontology based integration of IEC TC 57 standards. In: STASIS/BREIN workshop held during the 4th international conference on interoperability for enterprise software and applications (I-ESA ’08).” Berlin, Germany: I-ESA ’08, 2008.

Usman, Z., Young R.I.M., N. Chungoora, C. Palmer, K. Case and J.A. Harding. “Towards a formal manufacturing reference ontology.” *International Journal of Production Research* 51, no. 22 (2013): 6553–6572.

Usman, Z. *A Manufacturing Core Concepts Ontology to Support Knowledge Sharing, PhD Thesis.* Loughborough: Loughborough University, 2012.

Usman, Z. , Young R.I.M., N. Chungoora, C. Palmer, K. Case and J.A. Harding. “A Manufacturing Core Concepts Ontology for Product Life Cycle Interoperability.” *In International Ifip Working conference on Enterprise Interoperability.* Stockholm, Sweden: Springer, 2011.

Vrandečić., Denny. “Ontology Evaluation.” In *Handbook on Ontologies*. Springer, 2009.

Vujasinovic, M., Ivezic, N., Kulvatunyou, B. and Barkmeyer, E.,. *An industrial validation of a semantic-mediation architecture.* USA: National Institute of Standards and Technology. [Online] Available at:, 2007.

W3C. *OWL Web Ontology Language.* 2004. https://www.w3.org/TR/2004/REC-owl-guide-20040210/#Namespaces (accessed 2018).

W3C. *OWL Web Ontology Language Overview (D. L. McGuinness, and F. v. Harmelen, Editors).* 2004. http://www.w3.org/TR/owl-features/ (accessed 11 2017).

W3C. *OWL Web Ontology Language.* W3C, 2004.

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hubner, S. “Ontology-Based Integration of Information-A Survey of Existing Approaches.” *JCAI--01 Workshop: Ontologies and Information Sharing.* Retrieved from http://www.citeulike.org/user/Salma/article/593500, 2001.

Wang, F., Fenves, S.J., Sudarsan, R. and Sriram, R.D.,. “Towards modelling the evolution of product families.” *In: Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Chicago, Illinois, USA, 2003.

Wang, K., Tong, S. “An Ontology of Manufacturing Knowledge for Decision Support .” *Wireless Communications, Networking and Mobile Computing*, 2008.

Wang, Ming-Tzong, Marcella A. Chamberlain, Ajay Joneja and Tien-Chien Chang. “Manufacturing feature extraction and machined volume decomposition in a computer-integrated feature-based design and manufacturing planning environment.” *Computers in Industry* 23, no. 1-2 (1993): 75-86.

Wiig, K.M. *Knowledge Management Foundations: Thinking About Thinking - How People and Organisations Create, Represent and use Knowledge.* Arlington, TX: Schema Press, 1994.

Williams, T.J.,. “The Purdue Enterprise Reference Architecture.” *Computers in Industry* 24, no. 2-3 (1994): 141-158.

Wordnet. “Princeton University "About WordNet".” *WordNet. Princeton University.* 2010. http://wordnet.princeton.edu>.

Yang, B., Qiao L., Zhu Z., Wulan M. “A Metamodel for the Manufacturing Process Information Modeling.” *Procedia CIRP* , no. 56 (2016): 332-337.

Yang, Q.Z. and Zhang, Y.,. “Semantic interoperability in building design: methods and tools.” *Computer Aided Design* 38 (2006): 1099-1112.

Yao, Z., Bradley H.D, Marapoulos P.G,. “A concurrent engineering approach for supporting weld product design at early stages of the design process.” In *Artificial Intelligence in Design* , by F. Sudweeks J.S. Gero, 641-660. Netherlands: Kluwer Academic Publishers, 1998.

Ye, Y., Yang, D., Jiang, Z. and Tong, L. “Ontology-based semantic models for supply chain management.” *International Journal of Advanced Manufacturing Technology.* 37 (2008): 1250-1260.

Young, R.I.M.,Gunendran, A. G.,Cutting-Decelle, A.-F, Gruninger, M., “Manufacturing knowledge sharing in PLM: a progression towards the use of heavyweight.” *International Journal of Production Research* 45, no. 7 (2007): 1505–1519.

Young, B., Cutting-Decelle, A.-F, Guerra, D., Das, B., Cochrane, S. “Sharing Manufactruing Information and Knowledge in Design Decision and Support.” In *Advances in Integrated Design and Manufacturing in Mechanical Engineering*, by D. Brissoud, D. Coutellier, C. McMahon A. Bramley, 173-185. Springer, 2005.

Young, R.I.M. and Bell, R.,. “Design by features - advantages and limitations in machine planning integration.” *International Journal of Computer Integrated Manufacturing* 6, no. 1-2 (1993): 105-112.

Young., R, Chungoora N, Usman Z, Anjum N, Gunendran G, Palmer C, Cutting-Decelle A F,. “An Exploration of Foundation Ontologies and Verification Methods for Manufacturing Knowledge Sharing.” *International Conference on Interoperability of Enterprise Software and Applications:Proceedings of the Workshops and Doctoral Symposium of the I-ESA International Conference.* Wiley and Sons Inc, 2010. 83-93.

Zachman. *The Zachman Enterprise Framework.* http://www.zachmaninternational.com/, 2009.

Zaletelj, V, Vrabič R, Hozdić E, Butala P,. “A foundational ontology for the modelling of manufacturing systems.” *Advanced Engineering Informatics* 38 (2018): 129-141.

Zhao, J., Cheung, W.M. and Young, R.I.M.,. “A consistent manufacturing data model to support virtual enterprise.” *International Journal of Agile Management Systems.* 1, no. 3 (1999): 150-158.

Zhou, Jiehan and Rose Dieng-Kuntz. “Manufacturing Ontology Analysis and Design: Towards Excellent Manufacturing.” *IEEEXplore*, 2004: 39-45.

Zhu, Hongwei and Stuart E Madnick. *A Lightweight Ontology Approach to Scalable Interoperability.* Massachusetts,USA: In Working paper- Massachusetts Institute of Technology (MIT), Sloan School of Management, 2007.

Zhu., Z, Zhou X, Shao K,. “A novel approach based on Neo4j for multi-constrained flexible job shopscheduling problem.” *Computers and Industrial Engineering*, no. 130 (2019): 671-686.

Zhuoyu., H, Jowers C, Dehghan-Manshadi A, Dargusch M,. “Smart manufacturing and DVSM based on an Ontological approach.” *Computers in Industry* 117 (2020): 103189.

Appendices

A. Methodologies for Ontology Development

Blomqvist and Ohgren Methodology

(Blomqvist, 2008) established their own methodology after reviewing different ontology development methodologies. The 3 fundamental steps of their methodology were

a) Requirement Analysis: The very first step of the methodology is to identify the requirements of the ontology. It results in a document summarising the end users and the uses, scope and purpose of the ontology. Also, the functionality of the ontology and the required resources i.e the ontology languages, tools are incorporated in this document. This document is rigorously reviewed to eradicate any contradictions by agreeing on the semantics of the terms.

b) Ontology Construction: This is the crucial step within this methodology and describes the construction process of the ontology. The construction of the ontology is further divided into automatic and manual approach. Both of these approaches comprise of a step to build the ontology followed an implementation. However, the automatic approach has an additional step of reusing existing ontologies. The fundamental difference between the two approaches is the method of ontology creation.

   In the automatic approach, the ontology structure is created using software tools from the requirements documents and existing ontologies. More than one structure can be created using this approach, which can be implemented as a whole or partially. The different concepts and their relations are identified from the requirement documents using software tools (e.g. Text-to-onto (Maedche, 2003)). These are then matched for patterns automatically (Cohen, 2003).

   For the manual approach, the ontology structure is created manually. This encompasses the manual identification of the concepts and their relations. The constraints on the meanings of the concepts and the relations are established after the initial hierarchies are created.

c) Evaluation and Testing: The final ontology is built on a pattern which is better suitable to fulfil all the requirements and provide a comprehensive structure. The developed ontology is then tested on different systems.

The authors reported that automatic ontologies provide lesser details but are more compatible. Therefore, they proposed a hybrid approach. However, each approach had its own merit with regards to the context of use. The manual approach has been found to be more suited for developing core ontologies and for explicitly defining the constraints on the concepts (Usman, 2012).

Noy and McGuinness Methodology

The methodology proposed by (McGuinness, 2002) comprised of the following 7 fundamental steps

1. Determine the Domain of the Ontology: The first step is to establish the domain and scope of the ontology. Further, this step would
   a. Identify the end of users of the ontology
   b. Identify a set of competency questions that the ontology is required to answer.
   c. Identify the purpose of the ontology
2. Consider Reusing existing Ontology: The second step suggests the use of existing ontologies, such as importable digital ontologies.
3. Enumerate Important Term: This step entails the listing of all the important terms which are relevant to the concerned domain.
4. Define Classes and their Hierarchy: The process of defining the classes and their hierarchy can be either done using a top-down or a bottom up approach.
5. Define Properties and Attributes of the Classes: This step defines the different properties of the classes that their attributes. E.g. extrinsic properties define the external attributes, while the intrinsic defines the internal.

6. Define the facets of the slots/relations: This step establishes the cardinality (one to one, one to many etc.) of the relation. It also incorporates more information regarding the relations such as their domains, range and their type (e.g. String, Boolean etc.).
7. Create Instances: The final step of the ontology is to create instances of the defined classes.

This methodology suits the most of the ontology development requirements and is quite complete. However, an additional step to define the axioms or constraints on the concepts would enable the development of heavyweight ontologies.

B. Languages for Ontology Development

Categories of Ontology Development Languages	Languages	Short Description	Formalisation	Developer
Ontology Schematic Languages	IDEF 5	Integrated Definition schematic languages which are used to develop ontologies	Lightweight	KBSI Inc.
	UML & UML-2	Unified Modelling Language(UML) is an object oriented modelling language in the form of graphical notation.	Lightweight	Object Management Group
	EXPRESS-G	These are diagrammatic notations for modelling information	Lightweight	ISO 10303
Ontology Mark-up languages	XML	eXtensible Markup Language(XML) are syntaxes for a number of semantic web languages	Lightweight	World Wide Web (W3C)
	RDF, RDF(S)	Resource Description Framework (RDF) & Resource Description Framework Scheme (RDF-S) is used to model web resources information.	Lightweight	World Wide Web (W3C)
	OIL	Ontology Inference Layer (OIL) is an extension of RDF(S) with additional Knowledge Representation (KR) primitives.	Lightweight	Horrocks et al.
	DAML+OIL	Defence Advanced Research Projects Agency Markup Language (DAML) along with OIL is an extension of RDF(S). It is an semantic markup language for Web resources	Lightweight	Horrocks & van Harmelen, 2001
	OWL	Web Ontology Language (OWL) was developed from DAML+OIL and based on RDF(S).	Heavyweight	World Wide Web (W3C)
General Ontology Languages	OCML	Operational Conceptual Modelling Language (OCML) are the ones that supports 1st and 2nd order axioms.	Heavyweight	The Open University, Knowledge Media Institute
	EXPRESS	This language is primarily used for modelling product data	Lightweight	ISO 10303
	F-logic	An object oriented method for first order logic(FOL). It is used for object-oriented and deductive databases.	Heavyweight	Mannheim & Stony Brook University
	KIF	Knowledge Interchange Format (KIF) is a computer understandable format for FOL	Heavyweight	DARPA
	CL	Common Logic(CL) encompasses different logic based languages	Heavyweight	ISO/IEC 24707
	CLIF	Common Logic Interchange Format (CLIF) is similar to KIF for CL	Heavyweight	ISO/IEC 24707
	CGIF	Conceptual Graph Interchange Format (CGIF) is used to represent conceptual graphs in CL.	Heavyweight	ISO/IEC 24707
	KFL	Knowledge Framework Language (KFL) is based on ECLIF which is an extended part of CLIF.	Heavyweight	Highfleet Inc.

Unified Modelling Language (UML, UML-2)

Unified Modelling Language (UML) was developed by (Rumbaugh J, 1998) as a measure to standardise the different symbolic representation systems used in software design. It was initially developed for software design.

UML uses various diagrams for representations such as Communication Diagrams, Use-Case Diagrams and Class Diagrams. Class diagrams have been used in this research work which are the most commonly used diagrams. In UML, the classes are represented with boxes, which are divided into three parts describing the name, attributes and the operations. For the ontological community, operations are not used. The types of attributes and their values are expressed in their descriptions with cardinalities defined on them. However, UML have been found to be lacking in expressivity and are highly dependent on the platform (Usman, 2012). (Nitishal Chungoora, 2012) suggested the use of UML-2. This has the potential to represent ternary and higher relations and suites the lightweight representation for developing heavyweight ontology.

Web Ontology Language (OWL)

The W3C Web Ontology (WebOnt) Working Group identified several use-cases for ontologies on the Web requiring more expressiveness than RDF and RDFS (Staab, 2009). Hence, Web Ontology Language (OWL) was developed by combining DAML and OIL, while building upon RDF(S). The drawbacks of RDF and RDF(S), such as inability to represent constraint cardinality, special relations, disjoint classes and limiting the relations to certain number of classes have been addressed during development of OWL (Antoniou, 2009).

OWL has the capability to process the content of the information rather than just presenting it. This is provided by the additional richer vocabulary with formal semantics (W3C, 2004). OWL has been considered to have a high expressive power that can support various reasoning languages (Sengupta, 2013). It provides more machine interpretability than XML, RDF and RDF(S) (Usman, 2012). The formalism of OWL semantics is based on Open World Assumption (OWA) which basically works on the assumptions that things which are not known to be true, does not necessarily have to be false (Sengupta, 2013) (Nitishal Chungoora, 2012) (Sirin et al, 2008).

OWL is further classified into OWL Lite, OWL DL (Description Logic) and OWL Full. OWL Lite is primarily used for applications which require hierarchical classifications and simple constraints. OWL DL has an advantage over the others as it can provide all the results with maximum expressivity and within a limited computational time. Further it is one of the widely used heavyweight Description Logic based language. OWL Full utilises all the constructs of the OWL language and has maximum expressivity. However, it takes a longer computational time and lacks a support tool. OWL has also been acknowledged by the ISO standards community for their contribution in SemanticStep of the S-TEN project (S-TEN Project, 2011) and in the integration of the International Electrotechnical Commission (IEC) TC 57 standards (Uslar, 2008).

C. Tools for Ontology Development

Formalisation	Ontology Development Tools	Short Description	Developer
Lightweight	Enterprise Architect	Enterprise Architect is a software tool for visual modelling. It is a design tool based on Object Management Group (OMG) UML which supports the design and development of software systems , business process modelling and industry based domains,. EA is further used to develop lightweight UML model of the ontology. Its capability to represent ternary and higher order relations is its key advantage.	Sparx Systems
	Microsoft(MS) Visio	MS Visio is a tool for diagrammatic representation of various applications and supports UML and IDEF-5. This tool is only capable of representing the concepts and relations in the form if diagrams without any semantics.	Microsoft
	UML Studio	Unified Modelling Language (UML) Studio is a widely used UML modelling tool which can be used to develop lightweight ontologies. It includes industry standard notations and are capable for handling very large models.. It further supports ternary and higher order relations. This tool is used in this research for lightweight modelling.	Pragsoft Corporation
Heavyweight	Ontolingua Server	Developed by Stanford University to provide a distributed collaborative environment for browsing, creating, editing, modifying and using ontologies. However, it lacks an inference engine but support the translation of the ontologies into representation languages such KIF, CLIPS. LOOM , IDL, Prolog etc.	Knowledge Systems Lab, Stanford University
	OillEd	As the name suggests, OilEd was developed as an editor for OIL, DAML+OIL ontologies. This editor uses the DL approach and has consistency checking, classification functions. It uses the FaCT and RACER inference engines.	
	WebODE	WebODE is an integrated workbench for ontology representation, reasoning and exchange. It also facilitates integration of ontologies into Semantic Web applications and rapid development of applications using ontologies. The ontologies are created through an interactive approach in a three tier architecture of (a) ontology environment, (b) application server and (c) database management. The client server architecture helps in the utilisation of available resources and uses OKBC based inference engine. It is capable of importing, exporting and translation of RDF(S), OIL, DAML+OIL and F-Logic.	Technical University of Madrid
	OntoEdit	OntoEdit is a collaborative ontology development tool for the semantic web. It is a graphical editor which uses management structure of the Karlsruhe Institute Ontology (KAON) and is based on the CommonKADS, On-To Knowledge Methodologies. Ontobroker is used as the knowledge base by this editor and has inference capability. It is capable of importing and exporting DAML+OIL, RDF and F-Logic.	Applied Informatics & Formal Description Methods (AIFB), Karlsruhe University
	IODE	Integrated Ontology Development Environment (IODE) was developed by Highfleet as the only available commercial tool for CL based ontologies. eXtensible Knowledge Server (XKS) is used as the database server that supports the built ontologies. It uses the simplified version of CL called KFL and allows to write the code outside the development environment. This is imported into IODE which is finally loaded into XKS to be instantiated.	Highfleet Inc.
	Protégé	It is the most widely used ontology development tools supported by the development languages RDFS & OWL. More details are provided in the main section..	Knowledge Systems Lab, Stanford University

Protégé

Protégé was developed by the Stanford Medical Informatics (SMI) group of Stanford University. It was initially developed to aid expert systems in knowledge acquisition and it simplification. It is an open source software which is used to develop ontologies for domain models and knowledge bases (http://protege.stanford.edu/overview/). Protégé is a Java-based standalone application (Corcho, 2002) with the ontology editor as its core. It is widely used for ontology development due to widely available support (Khondoker, 2010).

The architecture of Protégé is extensible for creating and integrating newer extensions with other applications, tools, and knowledge bases. It supports ontology representation languages like OWL and RDFS (Gasevic, 2006). Further the user interface is customizable and the output file format is adaptable with multiple languages (FLogic, Jess, OIL, XML, Prolog) (Mizoguchi, 2009). The knowledge model of Protégé is based on frames and first order logic (Go´mez-Pe´rez, 2004). Its main modelling components are classes, slots, facets and instances. Protégé's knowledge model supports the expression of metaclasses (classes whose instances are also classes) and ontology merging. The capability of Protégé to integrate with Semantic Web Rule Language (SWRL) makes the ontology more rigorous and is its most beneficial aspect for this research.

In this research work, the Protégé-OWL ontology editor is used to develop and deploy the ontology. SWRL is used to define rules and axioms that provide the semantic rigor to the model. SWRL is an extension of OWL that permits complex rule definitions and advance reasoning over the concepts. The syntax followed in defining the rules are in the form of antecedent-consequent pairs. These rules help the system to interpret and infer the new knowledge. Furthermore, the Semantic Query-Enhanced Web Rule Language (SQWRL) is used to enquire the knowledge base for displaying the particular set of results. SQWRL is a query language for OWL which is based on SWRL. It is more concise, readable and semantically robust than other query languages such as SPARQL etc. (Connor, 2009).

D. Formalisation of PLO

<owl:Ontologyrdf:about="http://www.owl-ontologies/CoreDomain.org">

<owl:importsrdf:resource="http://www.owl-ontologies/ProductLifecycleCore.org"/>

</owl:Ontology>

<owl:Class rdf:about="http://www.owl-ontologies/PLOClasses.org#ProductLifecycleCore:Product">

<rdfs:subClassOf rdf:resource="http://www.owl-ontologies/Foundation.org#Foundation:PhysicalEndurant"/>

</owl:Class>

<owl:Class rdf:about="http://www.owl-ontologies/PLO.org#ProductLifecycleCore:DiscreteProduct">

<rdfs:subClassOf rdf:resource="http://www.owl-ontologies/PLO.org#ProductLifecycleCore:Product"/>

</owl:Class>

```
<owl:Class rdf:about="http://www.owl-
ontologies/PLO.org#ProductLifecycleCore:CompoundProduct">
<rdfs:subClassOf rdf:resource="http://www.owl-ontologies/PLO.org#
ProductLifecycleCore:DiscreteProduct"/>
</owl:Class>
<owl:Class rdf:about="http://www.owl-ontologies/PLO.org#
ProductLifecycleCore:AtomicProduct">
<rdfs:subClassOf rdf:resource="http://www.owl-ontologies/PLO.org#
ProductLifecycleCore:DiscreteProduct"/>
</owl:Class>
```

The following syntax is used to declare the object property relation *hasPart* followed by the declaration of *associatedTo* relation.

```
<owl:ObjectProperty rdf:about="http://www.owl-ontologies/PLO.org#hasPart">
<rdfs:domain rdf:resource="http://www.owl-ontologies/PLO.org# ProductLifecycleCore:Product"/>
<rdfs:range rdf:resource="http://www.owl-ontologies/PLO.org# ProductLifecycleCore:Part"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://www.owl-ontologies/PLO.org#associatedTo">
<rdfs:domain rdf:resource="http://www.owl-ontologies/PLO.org#
ProductLifecycleCore:ProductFeature"/>
<rdfs:range rdf:resource="http://www.owl-ontologies/PLO.org#ProductLifecycleCore:Product"/>
</owl:ObjectProperty>
```

The data property relations *hasNumberOfParts* and *hasDimension* are declared using the following syntax.

```
<owl:DatatypeProperty rdf:about="http://www.owl-ontologies/PLO.org#hasNumberOfParts">
<rdfs:domain rdf:resource="http://www.owl-ontologies/PLO.org# ProductLifecycleCore:Product"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="http://www.owl-ontologies/PLO.org#hasDimension">
<rdfs:domain rdf:resource="http://www.owl-ontologies/PLO.org#ProductLifecycleCore:Parameter"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>

<owl:Class rdf:about="http://www.owl-ontologies/PLO.org# DomainCoreGeneric:LiquidStateJoining">
<rdfs:subClassOf rdf:resource="http://www.owl-ontologies/PLO.org#
DomainCoreGeneric:AssemblyProcess"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.owl-ontologies/PLO.org#reachesMeltingTemperature"/>
<owl:someValuesFromrdf:resource="http://www.owl-
ontologies/PLO.org#ProductLifecycleCore:MeltingTemperature"/>
```

The following statement introduces a consistency checking for the instances of *Welding* by making sure that it has a relation with instances of *Part* through the *requiresPart* relation. This also infers the classes having similar restrictions as equivalent class.

```
<owl:Class rdf:about="http://www.owl-ontologies/PLO.org#DomainCoreSpecific:Welding">
<owl:equivalentClass>
<owl:Class>
<owl:intersectionOf rdf:parseType="Collection">
<rdf:Description rdf:about="http://www.owl-ontologies/PLO.org#DomainCoreSpecific:Welding"/>
<owl:Restriction>
<owl:onProperty rdf:resource="http://www.owl-ontologies/PLO.org#requiresParts"/>
<owl:someValuesFrom rdf:resource="http://www.owl-
ontologies/PLO.org#ProductLifecycleCore:Part"/>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>
```

The syntax for defining SWRL rule is shown below.

Product(?prod) ^ hasNumberOfParts(?prod, ?n) ^ swrlb:greaterThan(?n,1) ^ hasPart(?prod,?part) -> CompoundProduct(?prod)

The above statement infers that when an instance of *Product* has more than one *Part* then it is classified as a *Compound Product*. The statement can also be viewed to ascertain that every *Compound Product* requires to have more than one *Part* defined. A combination of the following statements below helps the system to infer the different types of *Feature* asserted into the knowledge base. In the following syntax, system makes sure that any instance of *Feature* asserted with its *Form* is classed as a *Form Feature.* Further, it ascertains whether the *Form Feature* is *Product Feature.* Finally, the consequent statements determine if the asserted *Product Feature* is either a *Design Feature* or a *Manufacturing Feature.*

Feature(?f) ^ Form(?form) ^ hasAttributeOfInterest(?f, ?form) -> FormFeature(?f)

FormFeature(?f) ^ Product(?prod) ^ associatedTo(?f, ?prod) -> ProductFeature(?f)

ProductFeature(?f) ^ Function(?func) ^ hasFunction(?f, ?func) -> DesignFeature(?f)

ProductFeature(?f) ^ ManufacturingProcess(?manufproc) ^ hasManufacturingProcess(?f,?manufproc) -> ManufacturingFeature(?f)

E. Welding Standards

Table 21 Welding standards referring to ISO/CEN

Standard Bodies Referring To ISO/CEN			
Osterreichisches Normungsinstitut - Austrian Standards Institute (ASI)	Austria	TC 037	ONORM EN ISO 4063:2011, ONORM EN 1792:2003
Bureau de Normalisation/Bureau voor Normalisatie (NBN)	Belgium		NBN CEN/TR 14599
Bulgarian Institute for	Bulgaria	TC-30	БДС EN 14610:2009

Standardization (BDS)			
Croatian Standards Institute (HZN)	Croatia		Refers CEN
Cyprus Organization for Standardisation (CYS)	Cyprus		Refers CEN
Czech Office for Standards, Metrology and Testing (UNMZ)	Czech Republic	70	Refers CEN
Dansk Standard (DS)	Denmark	S-047	DS DS-handbog 106.2
Estonian Centre for Standardisation (EVS)	Estonia		Refers CEN
Finish Standards Association (SFS)	Finland		Refers CEN
Standardization Institute of the Republic of Macedonia (ISRM)	Former Yugoslav Republic of Macedonia	TC 39	MKC EN 14610: 2010
Association Française de Normalisation (AFNOR)	France		FD ISO/TR 25901-3:2017,
Deutsches Institut für Normung (DIN)	Germany		DIN 1910-100 (2008-02), DIN EN ISO 4063 (2011-03),
National Quality Infrastructure System (NQIS/ELOT)	Greece		Refers to CEN
Hungarian Standards Institution (MSZT)	Hungary		Refers to CEN
Icelandic Standards (IST)	Iceland		Refers to CEN
National Standards Authority of Ireland (NSAI)	Ireland		I.S. EN ISO 4063:2010, I.S. CEN/TR 14599:2005, I.S. EN ISO 17659:2004
Ente Nazionale Italiano di	Italy		UNI CEN/TR 14599:2012,

Unificazione (UNI)			UNI EN ISO 17659:2006
Latvian Standard Ltd. (LVS)	Latvia		Refers to CEN
Lithuanian Standards Board (LST)	Lithuania	TK 41	LST CEN / TR 14599: 2013
Organisme Luxembourgeois de Normalisation (ILNAS)	Luxembourg		Refers to CEN
The Malta Competition and Consumer Affairs Authority (MCCAA)	Malta		Refers to CEN
Nederlands Normalisatie-instituut (NEN)	Netherlands		NEN NPR ISO/TR 25901-3:2016, NEN NPR ISO/TR 25901-4:2016, NEN NPR ISO/TR 25901-1:2016
Norges Standardiseringsforbund Standards Norway (SN)	Norway		NS EN ISO 4063:2010, NS EN 1792
Polish Committee for Standardization (PKN)	Poland	TC 165	PN EN 1792:2010, PN EN ISO 17659:2008, PN EN 14610:2008
Instituto Português da Qualidade (IPQ)	Portugal	CT 019	Refers to CEN
Romanian Standards Association (ASRO)	Romania	CT 39	Refers to CEN
Institute for Standardization of Serbia (ISS)	Serbia	M044	SRPS CEN/TR 14599:2009
Slovak Office of Standards Metrology and Testing (UNMS)	Slovakia		Refers to CEN
Slovenian Institute for Standardization (SIST)	Slovenia	TRM	Refers to CEN
Asociación Española de Normalización (UNE)	Spain	CTN 14	UNE CEN/TR 14599:2006, UNE EN ISO 17659:2005, UNE

			EN ISO 4063:2010, UNE EN 14610:2006
Standardiserings-Kommissionen I Sverige - Swedish Standards Institute (SIS)	Sweden	TK 134	SS EN ISO 4063 Ed. 3 (2010), SS EN 14610 Ed. 1 (2005), SS EN ISO 17659 Ed. 1 (2005)
Schweizerische Normen-Vereinigung (SNV)	Switzerland		SNV DIN 8528-1:1973, SN EN ISO 4063:2011, SN EN ISO 17659:2004
Turkish Standards Institution (TSE)	Turkey		TSE TS 6261
Standards Australia and Standards New Zealand (AS/NZ)	Australia and New Zealand	WD-001	AS 2812-2005
Gosudarstvennyy standart) (GOST):Euro-Asian Council for Standardization, Metrology and Certification (EASC)	Russia, Belarus, Moldova, Kazakhstan, Azerbaijan, Armenia, Kyrgyzstan, Uzbekistan, Tajikistan, Georgia, Turkmenistan	EASC	GOST R ИСО 857-1-2009, GOST R ИСО 17659-2009,
Japanese Standards Association (JSA): Japanese Industrial Standards (JIS)	Japan		JIS Z 3000-1,JIS Z 3000-2,JIS Z 3000-3, JIS Z 3000-4, JIS Z 3000-6
Korean Standards Association (KSA)	Korea		KS B ISO 857-2:2013
South African Bureau of Standards (SABS)	South Africa	44	SANS 10044-1 Ed. 3 (2004/R2011), SANS 4063 Ed. 3 (2011),

Standardization Administration of the People's Republic of China (SAC)	People's Republic of China		SAC GB/T 3375-94

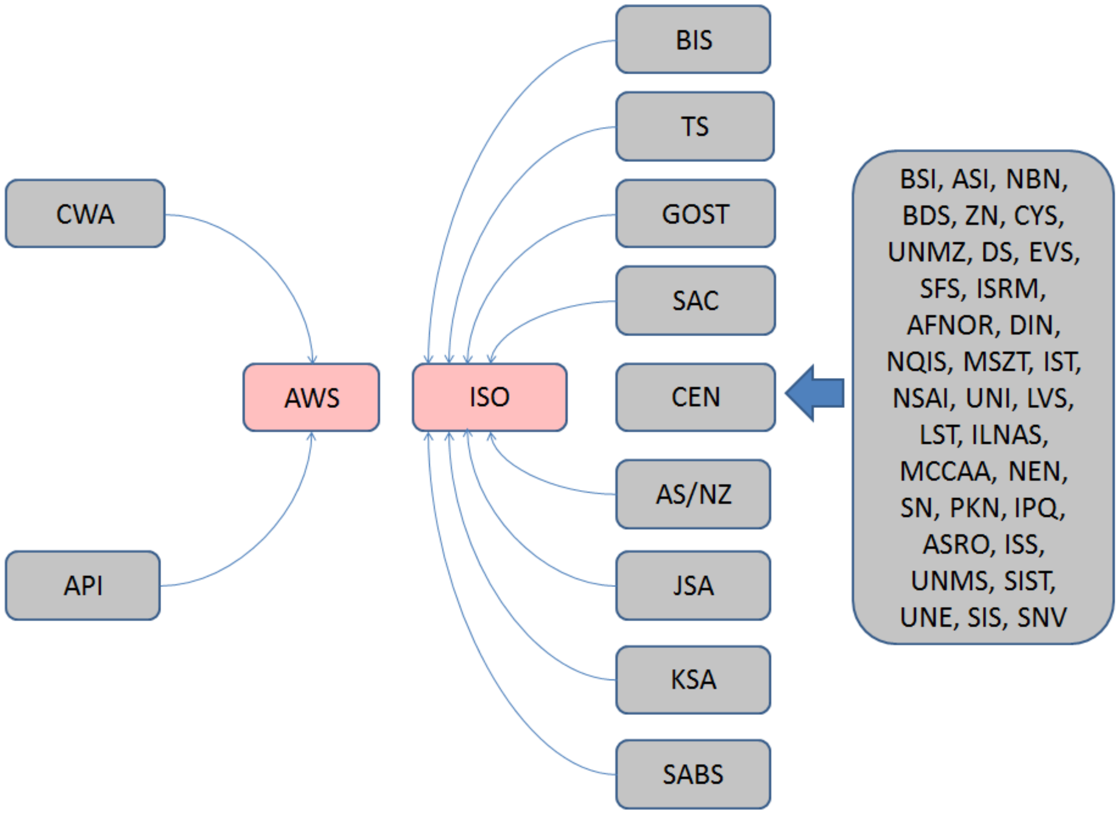


Figure 55 Map of welding standards

F. Rules and Axioms for Manufacturing Operations and Asserted Instances

(a) Rules and Axioms to Infer Manufacturing Operation Sequencing Knowledge

Table 22 Rules and Axioms to Infer Sequencing Knowledge

1	*ManufacturingOperation(?m1)^involvesOperatorIntervention(?m1,?o)^swrlb:stringEqualIgnoreCase(?o,"No")^ hasNumberOfOperatorIntervention(?m1, ?n) ^ swrlb:lessThan(?n, 1) ^ -> AutonomousOperation(?m1)* **Inference**: A *ManufacturingOperation* "m1" can be classified as an *AutonomousOperation* if it has "No" *OperatorIntervention* and the *NumberOfOperatorIntervention* is less than "1".
2	*ManufacturingOperation(?m2) ^ hasNumberOfOperatorIntervention(?m2, ?n) ^ involvesOperatorIntervention(?m2, ?o) ^ swrlb:stringEqualIgnoreCase(?o,"Yes") ^ swrlb:equal(?n, 1) -> SemiAutonomousOperation(?m2)* **Inference**: A *ManufacturingOperation* "m2" can be classified as a *SemiAutonomousOperation* if it has "Yes" *OperatorIntervention* and the *NumberOfOperatorIntervention* is equal to "1".
3	*ManufacturingOperation(?m3) ^ hasNumberOfOperatorIntervention(?m3, ?n) ^ involvesOperatorIntervention(?m3, ?o) ^ swrlb:stringEqualIgnoreCase(?o,"Yes") ^ swrlb:greaterThan(?n, 1) -> ManualOperation(?m3)* **Inference**: A *ManufacturingOperation* "m3" can be classified as a *ManualOperation* if it has "Yes" *OperatorIntervention* and the *NumberOfOperatorIntervention* is more than "1".
4	*AutonomousOperation(?op1) ^ ManualOperation(?op2) ^ SemiAutonomousOperation(?op3) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2) ^ hasPlatform(?op3, ?p3) ^ differentFrom(?p1, ?p2) ^ differentFrom(?p1, ?p3) ^ differentFrom(?p2, ?p3) -> canPrecedeOrFollow(?op1, ?op2) ^ canPrecedeOrFollow(?op1, ?op3) ^*

	canPrecedeOrFollow(?op2, ?op3) **Inference**: An *AutonomousOperation* "op1", *ManualOperation* "op2" and *SemiAutonomousOperation* "op3" *canPrecedeOrFollow* each other only if their respective *Platform*(s) are different.
5	*ManualOperation(?op1) ^ManualOperation(?op2) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2)^ hasSetup(?op1, ?s1) ^ hasSetup(?op2, ?s2) ^ hasResourceTool(?s1, ?t1) ^ hasResourceTool(?s2, ?t2) ^ differentFrom(?t1, ?t2) ^ differentFrom(?s1, ?s2) ^ sameAs(?p1, ?p2) -> canPrecedeOrFollow(?op1, ?op2)* **Inference**: A *ManualOperation* "op1" and another *ManualOperation* "op2" *canPrecedeOrFollow* each other only if they have the same *Platform* but different *Setup* because of different *ResourceTool*.
6	*SemiAutonomousOperation(?op1) ^ SemiAutonomousOperation(?op2) ^ hasSetup(?op1, ?s1) ^ hasResourceTool(?s1, ?t1) ^ hasResourceToolOrientation(?s1, ?to1) ^ hasWorkpieceOrientation(?s1, ?wpo1) ^ hasSetup(?op2, ?s2) ^ hasResourceTool(?s2, ?t2) ^ hasResourceToolOrientation(?s2, ?to2) ^ hasWorkpieceOrientation(?s2, ?wpo2) ^ differentFrom(?s1, ?s2) ^ sameAs(?t1, ?t2) ^ differentFrom(?to1, ?to2) ^ sameAs(?wpo1, ?wpo2) -> canPrecedeOrFollow(?op1, ?op2)* *SemiAutonomousOperation(?op1) ^ SemiAutonomousOperation(?op2) ^ hasSetup(?op1, ?s1) ^ hasResourceTool(?s1, ?t1) ^ hasResourceToolOrientation(?s1, ?to1) ^ hasWorkpieceOrientation(?s1, ?wpo1) ^ hasSetup(?op2, ?s2) ^ hasResourceTool(?s2, ?t2) ^ hasResourceToolOrientation(?s2, ?to2) ^ hasWorkpieceOrientation(?s2, ?wpo2) ^ differentFrom(?s1, ?s2) ^ sameAs(?t1, ?t2) ^ differentFrom(?to1, ?to2) ^ swrlb:notEqual(?wpo1, ?wpo2) -> canPrecedeOrFollow(?op1, ?op2)* *SemiAutonomousOperation(?op1) ^ SemiAutonomousOperation(?op2) ^ hasSetup(?op1, ?s1) ^*

	hasResourceTool(?s1, ?t1) ^ hasResourceToolOrientation(?s1, ?to1) ^ hasWorkpieceOrientation(?s1, ?wpo1) ^ hasSetup(?op2, ?s2) ^ hasResourceTool(?s2, ?t2) ^ hasResourceToolOrientation(?s2, ?to2) ^ hasWorkpieceOrientation(?s2, ?wpo2) ^ differentFrom(?s1, ?s2) ^ sameAs(?t1, ?t2) ^ sameAs(?to1, ?to2) ^ differentFrom(?wpo1, ?wpo2) -> canPrecedeOrFollow(?op1, ?op2) **Inference**: A *SemiAutonomousOperation* "op1" and another *SemiAutonomousOperation* "op2" *canPrecedeOrFollow* each other only if they have the same *Platform* but different *Setup* because of different *ResourceToolOrientation* or *WorkpieceOrientation* or both together.
7	*AutonomousOperation(?op1) ^ AutonomousOperation(?op2) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2) ^ involvesFeature(?op1, ?f1) ^ involvesFeature(?op2, ?f2) ^ isPrecededBy(?f2, ?f1) ^ differentFrom(?p1, ?p2) -> isFollowedBy1(?op1, ?op2)* **Inference**: An *AutonomousOperation* "op1" *isFollowedBy* a *AutonomousOperation* "op2" only if their respective *Platform*(s) are different and if the *Feature* "f2" involved in "op2" precedes the *Feature* "f1" involved in "op1".
8	*ManualOperation(?op1) ^ ManualOperation(?op2) ^ hasSetup(?op1, ?s1) ^ hasSetup(?op2, ?s2) ^ hasResourceTool(?s1, ?t1) ^ hasResourceTool(?s2, ?t2) ^ involvesFeature(?op1,?f1) ^ involvesFeature(?op2, ?f2) ^ isPrecededBy(?f2, ?f1) ^ differentFrom(?t1, ?t2) ^ -> isFollowedBy1(?op1, ?op2)* **Inference**: A *ManualOperation* "op1" *isFollowedBy* another *ManualOperation* "op2" only if they have the same *Platform* but different *Setup* because of different *ResourceTool* and if the *Feature* "f2" involved in "op2" precedes the *Feature* "f1" involved in "op1".
9	*AutonomousOperation(?op1) ^ SemiAutonomousOperation(?op2) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2) ^ involvesFeature(?op1, ?f1) ^ involvesFeature(?op2, ?f2) ^ isPrecededBy(?f2, ?f1) ^ differentFrom(?p1, ?p2) -> isFollowedBy1(?op1, ?op2)*

	Inference: An *AutonomousOperation* “op1” *isFollowedBy* a *SemiAutonomousOperation* “op2” only if their respective *Platform*(s) are different and if the *Feature* “f2” involved in “op2” precedes the *Feature* “f1” involved in “op1”. *SemiAutonomousOperation(?op1) ^ AutonomousOperation(?op2) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2) ^ involvesFeature(?op1, ?f1) ^ involvesFeature(?op2, ?f2) ^ isPrecededBy(?f2, ?f1) ^ differentFrom(?p1, ?p2) -> isFollowedBy1(?op1, ?op2)* **Inference**: A *SemiAutonomousOperation* “op1” *isFollowedBy* an *AutonomousOperation* “op2” only if their respective *Platform*(s) are different and if the *Feature* “f2” involved in “op2” precedes the *Feature* “f1” involved in “op1”.
10	*ManualOperation(?op1) ^ SemiAutonomousOperation(?op2) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2) ^ involvesFeature(?op1, ?f1) ^ involvesFeature(?op2, ?f2) ^ isPrecededBy(?f2, ?f1) ^ differentFrom(?p1, ?p2) -> isFollowedBy1(?op1, ?op2)* **Inference**: A *ManualOperation* “op1” *isFollowedBy* a *SemiAutonomousOperation* “op2” only if their respective *Platform*(s) are different and if the *Feature* “f2” involved in “op2” precedes the *Feature* “f1” involved in “op1”. *SemiAutonomousOperation(?op1) ^ ManualOperation(?op2) ^ hasPlatform(?op1, ?p1) ^ hasPlatform(?op2, ?p2) ^ involvesFeature(?op1, ?f1) ^ involvesFeature(?op2, ?f2) ^ isPrecededBy(?f2, ?f1) ^ differentFrom(?p1, ?p2) -> isFollowedBy1(?op1, ?op2)* **Inference**: A *SemiAutonomousOperation* “op1” *isFollowedBy* a *ManualOperation* “op2” only if their respective *Platform*(s) are different and if the *Feature* “f2” involved in “op2” precedes the *Feature* “f1” involved in “op1”.

11	*ManufacturingOperation(?op1) ^ ManufacturingOperation(?op2) ^ isFollowedBy1(?op1, ?op2) ^ requiresPrecedingFeatureStatus(?op2, ?s) ^ swrlb:lessThan(?s,1) -> isParallelWith(?op2, ?op1)* **Inference**: A *ManufacturingOperation* “op1” *isParallelWith* with *ManufacturingOperation* “op2” if the “op1” is followed by “op2” and the *requiresPrecedingFeatureStatus* “s” for “op2” is less than 1.

(b) Asserted Instances for Manufacturing Operation Sequencing Knowledge

Table 23 Asserted Instances for Manufacturing Operation Sequencing Knowledge

No	Operation	hasPlatform	hasMachineTool	hasFixture	hasSetup	hasTool	hasToolOrientation	hasWorkpieceOrientation
1	MachiningOP10	MachiningPlatform1	MachiningCentre1	MillingFixture1	MachiningSetup1	TurningCutter1	Horizontal	RestingOnBottomFace
2	MachiningOP20	MachiningPlatform2	MachiningCentre2	MillingFixture2	MachiningSetup2	TurningCutter1	Vertical	RestingOnTopFace
3	MachiningOP60	MachiningPlatform5	MachiningCentre1	TurningFixture1	MachininSetup3	TurningCutter1	Horizontal	RestingOnTopFace
4	MachiningOP70	MachiningPlatform6	MachiningCentre2	TurningFixture2	MachiningSetup4	TurningCutter1	Horizontal	RestingOnBottomFace
5	WeldingOP30	WeldingPlatform1	WeldingBode1	WeldingFixture1	WeldingSetup1	TackWeldingTorch	Horizontal	Vertical
6	WeldingOP40	WeldingPlatform1	WeldingBode2	WeldingFixture2	WeldingSetup2	FinalWeldingTorch	Horizontal	Vertical
7	InspectionOP50	InspectionPlatform1	CMM1	CMMFixture1	InspectionSetup1	CMMProbe1	Horizontal	RestingOnBottomFace
8	InspectionOP80	InspectionPlatform1	CMM2	CMMFixture2	InspectionSetup2	CMMProbe2	Vertical	RestingOnTopFace
9	MatIssueOP10	RawMaterialsStore	N/A	N/A	RawMaterialsStore	NA	N/A	N/A
10	FittingOP50	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	FittingBondingSetup1	FTLAssemblyTools,FAPAssemblyTools, TorqueControlledPowerTools	Variable	RestingOnTopFace
11	BondingOP60	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	FittingBondingSetup2	BlankingTools, GapFillingTools	Variable	RestingOnTopFace
12	InspectOP70	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	FittingBondingSetup3	Borescope	Variable	RestingOnTopFace
13	FittingOP80	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	FittingBondingSetup4	RALAssemblyTools, RAPAssemblyTools	Variable	RestingOnBottomFace
14	FittingOP90	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	FittingBondingSetup5	RearCaseAssemblyTools, MountRingAssemblyTools, OGVAssemblyTools, FrontCaseAssemblyTools	Variable	RestingOnBottomFace
15	BondingOP100	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	FittingBondingSetup6	InfillingTools	Variable	RestingOnBottomFace
16	InspectOP110	FittingBondingPlatform1	FittingBondingBay1	FittingBondingFixture1	InspectionSetup3	InspectionTools	Variable	RestingOnBottomFace

Table 24 List of Manufacturing Operations with their Operator Involvement and Requirement of Preceding Feature

Manufacturing Operation	**Involves Operator Intervention**	**Has Number Of Operator Intervention**	**Requires Preceding Feature Status**
MachiningOP10	No	0	1
MachiningOP20	Yes	1	1
MachiningOP60	Yes	1	1
MachiningOP70	Yes	1	1
WeldingOP30	Yes	3	1
WeldingOP40	Yes	3	1
InspectionOP50	No	0	1
MatIssueOP10	Yes	3	1
FittingOP50	Yes	2	0
BondingOP60	Yes	2	0
InspectOP70	Yes	2	1
FittingOP80	Yes	2	0
FittingOP90	Yes	2	0
BondingOP100	Yes	2	0
InspectOP110	Yes	2	0

Table 25 List of Operations and Features Worked On

Operation No.	**Operation**	**Feature Worked On**
Machining OP 10	Inner Ring Machining	Platform
Machining OP 20	Inner Ring Platform Machining	Stub
Machining OP 60	Stub Machining Operation	Knife Edge
Machining OP 70	Stub Face Machining Operation	Abutment Face
Inspection OP 50	Joint Inspection Operation	Position

Joining OP 30	Tack Welding Operation	Joint Locater
Joining OP 40	Welding Operation	Joint
Material Issue OP 10	Material Issue	Material Issue
Fitting OP 50	Front Fan Case Fitting Operation	Fan Track Liners, Front Acoustic Panels
Bonding OP 60	Front Fan Case Bonding Operation	Gap Fill Front Fan Case
Inspection OP 70	Front Fan Case Inspection Operation	Gap Cavities
Fitting OP 80	Rear Case Fitting Operation	Rear Acoustic Liners, Rear Acoustic Panels
Fitting OP 90	Fan Case Assembly Operation	Assembly of Front Case, OGV, Mount Ring and Rear Case
Bonding OP 100	Fan Case Bonding Operation	Infill Panels
Inspection OP 110	Fan Case Inspection Operation	All Fan Case Assembly Features

(c) Rules and Axioms Used in the Case Study

1	*swrlb:equal(?n, 1) ^ hasPart(?prod, ?part) ^ Product(?prod) ^ hasNumberOfParts(?prod, ?n) -> AtomicProduct(?prod)*
2	*hasPart(?prod, ?part) ^ swrlb:greaterThan(?n, 1) ^ Product(?prod) ^ hasNumberOfParts(?prod, ?n) -> CompoundProduct(?prod)*
3	*Form(?form) ^ hasForm(?f, ?form) ^ Feature(?f) -> FormFeature(?f)*
4	*FormFeature(?ff) ^ associatedTo(?ff, ?prod) ^ Product(?prod) -> ProductFeature(?ff)*
5	*Function(?func) ^ hasFunction(?prodf, ?func) ^ ProductFeature(?prodf) ->*

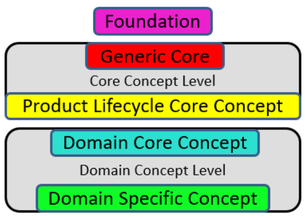
	DesignFeature(?prodf)
6	*hasManufacturingProcess(?prodf, ?manufproc) ^ ProductFeature(?prodf) ^ ManufacturingProcess(?manufproc) -> ManufacturingFeature(?prodf)*
7	*ManufacturingProcess(?mp) ^ JointType(?jtp) ^ MatingConfiguration(?mc) ^ producesJointType(?mp, ?jtp) ^ requiresMatingConfiguration(?mp, ?mc) -> AssemblyProcess(?mp)*
8	*decidesConformity(?mp, ?cf) ^ ManufacturingProcess(?mp) ^ Conformity(?cf) -> InspectionProcess(?mp)*
9	*ManufacturingFeature(?mf) ^ MachiningProcess(?mp) ^ hasManufacturingProcess(?mf, ?mp) -> MachiningFeature(?mf)*
10	*AssemblyProcess(?mp) ^ ManufacturingFeature(?mf) ^ hasManufacturingProcess(?mf, ?mp) -> AssemblyFeature(?mf)*
11	*ManufacturingFeature(?mf) ^ InspectionProcess(?mp) ^ hasManufacturingProcess(?mf, ?mp) -> InspectionFeature(?mf)*
12	*Camber(?c) ^ hasCamber(?tcl, ?c) ^ TangentialChordLength(?tcl) -> TangentialChordLengthWithCamber(?tcl)*
13	*swrlb:lessThan(?d2, 0.38) ^ hasParameter(?f, ?ccprof) ^ swrlb:lessThan(?d1, 0.38) ^ Form(?f) ^ hasDimension(?ccprof, ?d1) ^ hasForm(?mf, ?f) ^ hasDimension(?cxprof, ?d2) ^ ConcaveProfile(?ccprof) ^ swrlb:greaterThanOrEqual(?d2, 0.19) ^ swrlb:greaterThanOrEqual(?d1, 0.19) ^ TangentialChordLengthWithCamber(?tclwc) ^ hasParameter(?f, ?tclwc) ^ hasParameter(?f, ?dc) ^ ManufacturingFeature(?mf) ^ DatumCamber(?dc) ^ hasParameter(?f, ?cxprof) ^ hasDimension(?tclwc, ?d3) ^ ConvexProfile(?cxprof) ^ swrlb:lessThanOrEqual(?d3, 286.924) ^ hasCamber(?tclwc, ?dc) ^ swrlb:greaterThanOrEqual(?d3, 286.324) -> ManufacturableParameters(?cxprof) ^ ManufacturableFeatures(?mf) ^ ManufacturableParameters(?ccprof) ^ ManufacturableParameters(?tclwc)*

14	*hasForm(?mf, ?f) ^ hasParameter(?f, ?ccprof) ^ hasParameter(?f, ?cxprof) ^ ConvexProfile(?cxprof) ^ Form(?f) ^ hasParameter(?f, ?dc) ^ hasTolerance(?cxprof, ?t2) ^ swrlb:lessThanOrEqual(?t1, 0.03) ^ ManufacturingFeature(?mf) ^ swrlb:lessThanOrEqual(?t2, 0.03) ^ DatumCamber(?dc) ^ ConcaveProfile(?ccprof) ^ hasTolerance(?ccprof, ?t1) -> InspectableFeatures(?mf)*
15	*hasDimension(?dcafl, ?d4) ^ swrlb:lessThanOrEqual(?d2, 30.0) ^ swrlb:lessThanOrEqual(?d1, 30.0) ^ swrlb:greaterThanOrEqual(?d4, 7.0) ^ ConcaveKnifeEdgeAngle(?cckea) ^ Form(?f) ^ hasParameter(?f, ?cxkea) ^ ConvexKnifeEdgeAngle(?cxkea) ^ hasForm(?mf, ?f) ^ swrlb:greaterThanOrEqual(?d1, 0.0) ^ swrlb:greaterThanOrEqual(?d2, 0.0) ^ hasDimension(?cckea, ?d1) ^ hasParameter(?f, ?cckea) ^ swrlb:lessThanOrEqual(?d3, 1.0) ^ ManufacturingFeature(?mf) ^ swrlb:greaterThanOrEqual(?d3, 0.6) ^ Length(?dcafl) ^ hasParameter(?f, ?dckewph) ^ swrlb:lessThanOrEqual(?d4, 9.0) ^ hasParameter(?f, ?dc) ^ DatumCamber(?dc) ^ hasDimension(?dckewph, ?d3) ^ hasDimension(?cxkea, ?d2) ^ hasParameter(?f, ?dcafl) ^ DatumCamberStubHeight(?dckewph) -> ManufacturableParameters(?cxkea) ^ ManufacturableParameters(?dcafl) ^ ManufacturableParameters(?cckea) ^ ManufacturableFeatures(?mf) ^ ManufacturableParameters(?dckewph)*
16	*swrlb:greaterThanOrEqual(?d8, 7.65) ^ hasParameter(?f, ?rg) ^ hasParameter(?f, ?srft) ^ swrlb:greaterThanOrEqual(?d2, 3.83) ^ hasForm(?mf, ?f) ^ swrlb:lessThanOrEqual(?d9, 2.0) ^ StubDepthOfPreparation(?sdop) ^ OGVWeldPrepThickness(?owpt) ^ swrlb:lessThanOrEqual(?d2, 6.24) ^ swrlb:greaterThanOrEqual(?d9, 1.6) ^ swrlb:lessThanOrEqual(?d5, 60.00) ^ swrlb:greaterThanOrEqual(?d5, 50.00) ^ swrlb:lessThanOrEqual(?d1, 6.64) ^ hasParameter(?f, ?cxga) ^ StubWeldPrepThickness(?swpt) ^ hasDimension(?odop, ?d1) ^ hasDimension(?rg, ?d9) ^*

	hasParameter(?f, ?dc) ^ swrlb:equal(?d3, 0.0) ^ swrlb:equal(?d4, 0.0) ^ ManufacturingFeature(?mf) ^ hasParameter(?f, ?ccga) ^ swrlb:lessThanOrEqual(?d8, 13.03) ^ swrlb:lessThanOrEqual(?d7, 12.61) ^ hasParameter(?f, ?owpt) ^ ConvexSideGrooveAngle(?cxga) ^ swrlb:greaterThanOrEqual(?d7, 6.74) ^ hasParameter(?f, ?swpt) ^ hasDimension(?ccga, ?d6) ^ hasParameter(?f, ?sdop) ^ ConcaveSideGrooveAngle(?ccga) ^ swrlb:lessThanOrEqual(?d6, 50.00) ^ OGVRootFaceThickness(?orft) ^ swrlb:greaterThanOrEqual(?d6, 40.00) ^ hasDimension(?owpt, ?d7) ^ StubRootFaceThickness(?srft) ^ hasDimension(?orft, ?d3) ^ hasDimension(?srft, ?d4) ^ Form(?f) ^ hasParameter(?f, ?odop) ^ hasDimension(?cxga, ?d5) ^ RootGap(?rg) ^ hasParameter(?f, ?orft) ^ hasDimension(?sdop, ?d2) ^ DatumCamber(?dc) ^ OGVDepthOfPreparation(?odop) ^ swrlb:greaterThanOrEqual(?d1, 3.13) ^ hasDimension(?swpt, ?d8) -> ManufacturableFeatures(?mf)
17	*hasDimension(?ckea, ?d2) ^ ConcaveDiamondAngle(?cda) ^ hasDimension(?cda, ?d1) ^ swrlb:add(?d3, ?d1, ?d2) ^ ConcaveSideGrooveAngle(?ga) ^ ConcaveKnifeEdgeAngle(?ckea) -> hasDimension(?ga, ?d3)*
18	*ConvexSideGrooveAngle(?ga) ^ hasDimension(?ckea, ?d2) ^ ConvexKnifeEdgeAngle(?ckea) ^ ConvexDiamondAngle(?cda) ^ hasDimension(?cda, ?d1) ^ swrlb:add(?d3, ?d1, ?d2) -> hasDimension(?ga, ?d3)*
19	*ManufacturingFeature(?mf) ^ hasParameter(?f, ?ccprof) ^ Form(?f) ^ hasDimension(?ccprof, ?d1) ^ hasForm(?mf, ?f) ^ swrlb:greaterThan(?d1, 0.38) ^ ConcaveProfile(?ccprof) -> NonManufacturableFeatures(?mf) ^ NonManufacturableParameters(?ccprof)*
20	*swrlb:lessThan(?d1, 0.19) ^ ManufacturingFeature(?mf) ^ hasParameter(?f, ?ccprof) ^ Form(?f) ^ hasDimension(?ccprof, ?d1) ^ hasForm(?mf, ?f) ^ ConcaveProfile(?ccprof) -> NonManufacturableFeatures(?mf) ^ NonManufacturableParameters(?ccprof)*

21	*ManufacturingFeature(?mf) ^ hasParameter(?f, ?cxprof) ^ Form(?f) ^ hasForm(?mf, ?f) ^ swrlb:greaterThan(?d1, 0.38) ^ ConvexProfile(?cxprof) ^ hasDimension(?cxprof, ?d1) -> NonManufacturableFeatures(?mf) ^ NonManufacturableParameters(?cxprof)*
22	*swrlb:lessThan(?d1, 0.19) ^ ManufacturingFeature(?mf) ^ hasParameter(?f, ?cxprof) ^ Form(?f) ^ hasForm(?mf, ?f) ^ ConvexProfile(?cxprof) ^ hasDimension(?cxprof, ?d1) -> NonManufacturableFeatures(?mf) ^ NonManufacturableParameters(?cxprof)*
23	*TangentialChordLengthWithCamber(?tclwc) ^ hasParameter(?f, ?tclwc) ^ ManufacturingFeature(?mf) ^ DatumCamber(?dc) ^ Form(?f) ^ hasForm(?mf, ?f) ^ hasDimension(?tclwc, ?d) ^ swrlb:lessThanOrEqual(?d, 286.3) ^ hasCamber(?tclwc, ?dc) -> NonManufacturableFeatures(?mf) ^ NonManufacturableParameters(?tclwc)*
24	*TangentialChordLengthWithCamber(?tclwc) ^ hasParameter(?f, ?tclwc) ^ ManufacturingFeature(?mf) ^ DatumCamber(?dc) ^ Form(?f) ^ hasForm(?mf, ?f) ^ swrlb:greaterThanOrEqual(?d, 286.9) ^ hasDimension(?tclwc, ?d) ^ hasCamber(?tclwc, ?dc) -> NonManufacturableFeatures(?mf) ^ NonManufacturableParameters(?tclwc)*
25	*hasDimension(?dcsh, ?d2) ^ RootGap(?rg) ^ DatumCamberOGVHeight(?dcogvh) ^ swrlb:add(?d3, ?d1, ?d2) ^ DatumCamberStubHeight(?dcsh) ^ hasDimension(?dcogvh, ?d1) -> hasDimension(?rg, ?d3)*

G. PLO Concepts



** Concepts used for further exploitation, verification and newly proposed.*

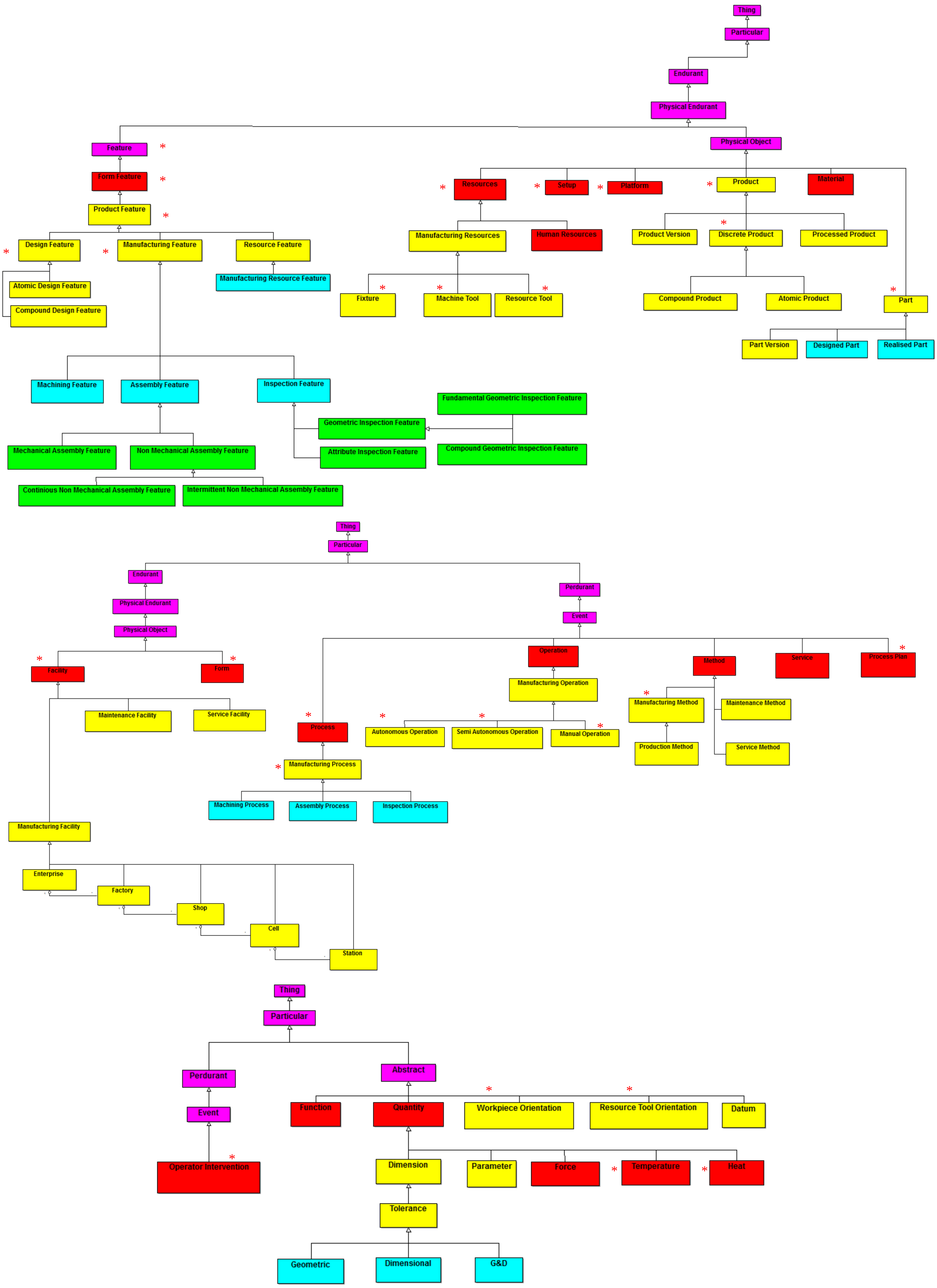


Figure 56Combined PLO with specialisation of concepts